

УДК 004.896

**Е.В. Дубиковская, Д.А. Гринюк, М.В. Чепурко, Д.О. Арпентий**

Белорусский государственный технологический университет

Минск, Беларусь

## **ВОЗМОЖНОСТИ ПОДДЕРЖКИ СЛОЖНЫХ ВЫЧИСЛЕНИЙ НА ПОЛЕВОМ УРОВНЕ СИСТЕМ УПРАВЛЕНИЯ**

*Аннотация.* Для повышения эффективности функционирования систем управления технологическими процессами существуют ограничения со стороны традиционных технических решений. Проведен анализ современных информационных технологий и вычислительных ресурсов технических средств автоматизация для расширения возможностей анализа и управления

**E.V. Dubikovskaya, D.A. Hryniuk, M.V. Chepurko, D.O. Arpenty**

Belarusian State Technological University

Minsk, Belarus

## **CAPABILITIES TO SUPPORT COMPLEX COMPUTING AT THE FIELD LEVEL OF CONTROL SYSTEMS**

*Abstract.* To improve the efficiency of process control systems, there are limitations from traditional technical solutions. An analysis of modern information technologies and computing resources of technical automation tools was carried out to expand the capabilities of analysis and management

Программируемые логические контроллеры (ПЛК) являются ключевым элементом в промышленной автоматизации, обеспечивая управление и контроль над различными производственными процессами. ПЛК представляют собой форму компьютеров, применяемых в промышленных системах управления и других областях, где необходимо осуществлять цифровое управление электрооборудованием. Они созданы для высокой прочности и надежности, а их программирование осуществляется при помощи специализированного программного обеспечения, позволяющего выполнять различные функции управления. В традиционной автоматизации часто применяются языки программирования, определенные стандартом МЭК 61131-3.

Однако принципы, заложенные в стандарте МЭК 61131-3, становятся недостаточно адаптированными для эффективного решения задач управления технологическими процессами. Концепция разработки современных ПЛК и их ключевой стратегии внедрения

алгоритмов управления также претерпевают изменения. Вместо прежних закрытых систем, на передний план выходит Linux, становясь основным стержнем в реализации данных алгоритмов.

Существует множество вариантов программных и аппаратных решений в области промышленной автоматизации, однако, использование Python предоставляет несколько значительных преимуществ.

Python — это высокоуровневый интерпретируемый язык программирования, который был создан в конце 1980-х годов Гвидо ван Россумом. Он известен своей простотой, элегантностью и читаемостью. Python является интерпретируемым языком, что означает, что он выполняется построчно с помощью интерпретатора, а не компилируется в машинный код. Это упрощает процесс разработки и отладки программ, так как программисты могут немедленно видеть результаты своего кода. Python нашел широкое применение в различных областях, включая научные исследования, разработку веб-приложений, автоматизацию, обработку данных, машинное обучение, искусственный интеллект, а также в сетевом и системном программировании.

Python поддерживает большинство операционных систем, включая Windows, macOS, и различные дистрибутивы Linux. В сравнении с другими языками, такими как C++, Python считается языком более высокого уровня из-за своего выразительного синтаксиса, который во многих случаях напоминает естественный язык, и обширного набора встроенных структур данных, таких как списки, кортежи, множества и словари. Python обладает огромным количеством библиотек и фреймворков, которые позволяют разработчикам эффективно решать разнообразные задачи. Что делает его универсальным языком программирования, а также привлекательным выбором для разработчиков, работающих в разных сферах.

Тем не менее, активный темп развития Python также влечет за собой свои трудности. Постоянные обновления создают проблемы совместимости между версиями. Python, как интерпретируемый язык высокого уровня, не может конкурировать с C++ по скорости и эффективности. Однако это преодолевается использованием библиотек, написанных на C++ и предварительно скомпилированных. В таких случаях Python выполняет роль скриптового языка, обеспечивая эффективную интеграцию с кодом, скомпилированным на C++.

Интеграция программируемых логических контроллеров и Python предоставляет широкий набор преимуществ для автоматизации. ПЛК являются надежными и приспособленными к трудным промышленным условиям устройствами, что делает их идеальным выбором для

управления электрооборудованием в различных секторах. Эта устойчивость позволяет создавать надежные системы управления, способные справляться с жесткими требованиями производства. В совокупности с широким спектром функциональных возможностей Python получаем мощный инструмент для разработки сложных алгоритмов управления и решения разнообразных задач в автоматизации.

Гибкость и универсальность Python выделяют его среди других языков программирования. Эта особенность облегчает обучение новичков в программировании и ускоряет адаптацию инженеров, не знакомых с языками МЭК 61131-3.

При всех преимуществах использование Python не всегда является оптимальным выбором для всех аспектов, связанных с программируемыми логическими контроллерами. Например, интеграция Python с специализированными аппаратными или программными системами может представлять трудности по сравнению с использованием других языков. Также в случаях, где требуются тесные связи с оборудованием или реальным временем ПЛК, более предпочтительными могут быть языки программирования, такие как релейная логика или структурированный текст. Необходимо объективно оценить все возможности языка программирования перед началом проекта.

Использование Python с программированными логическими контроллерами может быть целесообразным в различных случаях. Обычно лучше всего воспользоваться Python, когда требуется решение задач обработки высокого уровня. Это включает в себя проведение анализа данных, визуализацию и применение методов машинного обучения к данным, собранным с ПЛК.

На языке Python существует несколько специализированных библиотек, предназначенных для работы с ПЛК, такие как pyModbus или python-snap7. Также возможно использование библиотек общего назначения для последовательной связи, например, pySerial, для взаимодействия с ПЛК через последовательное соединение. Для передачи данных в и из ПЛК через сетевое соединение часто используют протоколы, такие как Modbus или Ethernet/IP.

Для взаимодействия с ПЛК и управления им можно использовать библиотеку pyModbus. Она предназначена для работы с протоколом Modbus, который широко используется в промышленной автоматизации для обмена данными между устройствами. PyModbus поддерживает как последовательный протокол Modbus RTU, так и сетевой протокол Modbus TCP. Это позволяет использовать библиотеку для

взаимодействия с устройствами, поддерживающими различные физические интерфейсы. Библиотека предоставляет обширный набор функций для чтения и записи различных типов данных, таких как биты, байты, 16-битные и 32-битные целые числа, вещественные числа и другие. Что делает pyModbus универсальным инструментом для обмена данными с различными устройствами.

Для работы с библиотекой pyModbus следует выполнить установку этого инструмента, а затем применить его для установки соединения по протоколу Modbus TCP с программным логическим контроллером (ПЛК). После успешного установления связи, доступны функции pyModbus для чтения и записи данных в регистры и обмотки ПЛК.

Многие факторы в настоящее время привлекают внимание к языку программирования Python. Прежде всего, Индустрия 4.0 представляет собой концепцию цифровой трансформации производства, в которой современные технологии информации и связи интегрируются в промышленные процессы, создавая умные, автоматизированные и взаимосвязанные системы производства. Что изменяет наше представление о промышленной автоматизации.

Еще одним важным аспектом стал промышленный Интернет вещей (IIoT), который представляет собой сеть физических устройств, оборудованных датчиками, программным обеспечением и другой технологией для сбора и обмена данными через Интернет. Это приводит к слиянию информационных технологий и операционных технологий, разрушая традиционные границы между специалистами в этих областях. В конечном итоге устройства IIoT используют полученную информацию для оптимизации своей работы.

Возвращаясь к Python, рассмотрим его сильные стороны в контексте промышленного интернета вещей (IIoT). Прежде всего, следует отметить способность Python эффективно обрабатывать обширные объемы данных. Во-вторых, Гвидо Ван Россум, создатель Python, разработал язык, сделав его высококочитаемым, что является ключевым фактором при совместной работе инженеров над кодом или его поддержке.

Особое влияние он оказывает в области машинного обучения (ML) и искусственного интеллекта (ИИ), где алгоритмы основаны на данных, а не на явно закодированных правилах. Применение Python включает в себя такие области, как профилактическое обслуживание и автономная робототехника. В данной сфере большинство современных инструментов машинного обучения, таких как PyTorch и TensorFlow, а также облачные сервисы, вроде AWS SageMaker от Amazon,

предоставляют встроенную поддержку для Python.

Запуская вычисления на периферийных устройствах с интегрированными графическими процессорами или используя ресурсы локального шлюза ПоТ для туманных вычислений, мы можем применять обученную модель на месте [1]. В области компьютерного зрения, необходимого для того, чтобы роботизированная рука могла что-то взять, машины используют свои камеры в роли глаз и способны распознавать объекты [1].

Несколько авторов рассматривают Python как альтернативу Matlab [2-4] и высоко ценят его преимущества. В конечном итоге, умение использовать Python для решения задач автоматического управления становится несомненным преимуществом для инженера по автоматизации.

### **Список использованных источников**

1. Jivan S. Parab, Madhusudan Ganuji Lanjewar, Marlon Darius Sequeira, Gourish Naik, Arman Yusuf Shaikh. Python Programming Recipes for IoT Applications, – Springer Singapore 2013 192 p.

2. Kumar, R.Mathusoothana & Lakshmi, S.L. & K.V., Shiny & P., Venkadesh. (2023). Problem Solving and Python Programming. 10.59646/pythonprog/049.

3. Raja, K. (2023). Python-based fuzzy logic in automatic washer control system. Soft Computing. 27. 1-27. 10.1007/s00500-023-07979-3.

4. Shaw, Rabindra. (2021). Innovations in Electrical and Electronic Engineering. 10.1007/978-981-16-0749-3\_.

УДК 621.798:664

**Е.А. Дынько, П.Г. Крупская**

Белорусский национальный технический университет  
Минск, Беларусь

### **ИНТЕЛЛЕКТУАЛЬНАЯ УПАКОВКА: ОБЗОР, ТЕНДЕНЦИИ, ИННОВАЦИИ**

*Аннотация.* Потребители заинтересованы в покупке безопасных продуктов питания. В связи с этим разработаны интеллектуальные упаковки, отслеживающие состояние продукта и предотвращающие покупку