

А. О. Фирсин, Е. А. Носова

МОДУЛЬНАЯ ПРОГРАММНАЯ АРХИТЕКТУРА ДЛЯ УПРАВЛЕНИЯ ШНЕКОВЫМ ЭКСТРУДЕРОМ В FGF-ПЕЧАТИ

Ключевые слова: аддитивное производство, программная архитектура, FGF-печать, шнековый экструдер, автоматизированное управление, модульность.

В работе представлена модульная программная архитектура для автоматизированного управления шнековым экструдером в аддитивных технологиях Fused Granular Fabrication (FGF). Актуальность исследования обусловлена расширением применения аддитивных технологий от прототипирования к созданию крупногабаритных изделий, где традиционные промышленные системы управления на базе программируемых логических контроллеров обладают закрытой архитектурой и ограничены в возможностях адаптации. Предложенная архитектура построена на принципах модульности, разделения ответственности и слабого зацепления компонентов, что достигается за счет централизованного координатора, через который осуществляется взаимодействие всех функциональных модулей. Такой подход реализует топологию «звезда» и гарантирует, что модули не вызывают методы друг друга напрямую, а взаимодействуют только через общий координатор. Архитектура предусматривает интеграцию пяти основных функциональных модулей: температурного контроля в четырех зонах экструдера (бункер и три нагревательные зоны), ведение журнала событий, управление подачей гранул, управление шнеком и видеомониторинг. Все динамические процессы реализуются в однопоточном режиме с использованием таймеров, что исключает необходимость в сложной синхронизации и гарантирует детерминированность поведения системы. Для сохранения состояния между сеансами работы применяется механизм постоянного хранения настроек. Предложенная архитектура ориентирована на поддержку принятия решений оператором и может служить основой как для замены, так и для расширения функционала существующих промышленных шкафов управления, включая перспективную интеграцию через стандартные протоколы. Архитектура предусматривает простую замену компонентов и добавление новых модулей, что делает ее пригодной для исследовательских и опытно-промышленных задач, а также для образовательных целей при изучении основ управления в FGF-печати.

А. О. Firsin, E. A. Nosova

MODULAR SOFTWARE ARCHITECTURE FOR SCREW EXTRUDER CONTROL IN FGF PRINTING

Keywords: additive manufacturing, software architecture, FGF printing, screw extruder, automated control, modularity.

This paper presents a modular software architecture for the automated control of a screw extruder in the Fused Granular Fabrication (FGF) additive technology. The relevance of the research is due to the expansion of the use of additive technologies from prototyping to the creation of large-sized products, where traditional industrial control systems based on programmable logic controllers have a closed architecture and limited adaptability. The proposed architecture is based on the principles of modularity, separation of responsibilities and loose coupling of components, which is achieved through a centralized coordinator through which all functional modules interact. This approach implements the "star" topology and ensures that modules do not call each other's methods directly, but interact only through a common coordinator. The architecture provides for the integration of five main functional modules: temperature control in four zones of the extruder (hopper and three heating zones), event logging, pellet feed control, auger control and video monitoring. All dynamic processes are implemented in single-threaded mode using timers, which eliminates the need for complex synchronization and guarantees deterministic system behavior. A permanent settings storage mechanism is used to save the state between work sessions. The proposed architecture is focused on supporting operator decision-making and can serve as a basis for both replacing and expanding the functionality of existing industrial control cabinets, including future integration through standard protocols. The architecture provides for a simple replacement of components and the addition of new modules, which makes it suitable for research and experimental industrial tasks, as well as for educational purposes when learning the basics of management in FGF printing.

В современной индустрии аддитивного производства наблюдается устойчивое расширение масштабов применения – от прототипирования к созданию крупногабаритных изделий [1-2]. В этом контексте технология Fused Granular Fabrication (FGF), основанная на послойном наплавлении гранулированных полимеров, занимает ключевую позицию, обладая значительными преимуществами в стоимости материала и скорости построения крупногабаритных заготовок [3-4]. В основе FGF-печати лежит использование шнекового экструдера, который обеспечивает подачу и плавление гранулированного сырья непо-

средственно перед нанесением на печатную платформу [5-7]. В таких условиях разработка программного обеспечения для управления экструдером становится важной задачей, поскольку точность и надежность системы напрямую влияют на качество изготавливаемых изделий [8-10].

Существующие промышленные системы управления, как правило, реализованы на базе программируемых логических контроллеров (ПЛК) и имеют «закрытую» архитектуру, что ограничивает возможности их адаптации для исследовательских задач, быстрого прототипирования новых алгоритмов управления и интеграции с современными системами

цифрового проектирования и управления жизненным циклом изделия. В связи с этим актуальной задачей является разработка гибких и масштабируемых программных решений, которые не только дублируют базовый функционал промышленных контроллеров, но и предоставляют широкие возможности для кастомизации, сбора данных и анализа процесса.

В данной работе предлагается подход к созданию такой системы на основе модульной программной архитектуры [11-13]. Целью исследования являлась разработка архитектуры программного обеспечения, обеспечивающей автоматизированное управление ключевыми параметрами в FGF-печати при активном участии оператора в цикле управления, а также обеспечивающей простоту дальнейшего расширения и

потенциальную интеграцию с физическим оборудованием.

Архитектура программного обеспечения разработана с использованием языка программирования, обеспечивающего кроссплатформенность и зрелость экосистемы [14-15]. Архитектурное решение предусматривает автоматизированное управление технологическим процессом, при котором оператор инициирует управляющие воздействия через элементы пользовательского интерфейса.

Архитектура приложения представляет собой композицию нескольких взаимосвязанных модулей, координируемых главным классом App (рис. 1). Данный класс отвечает за инициализацию главного окна (JFrame), создание всех компонентов и управление переключением между функциональными панелями.



Рис. 1 – Структура приложения
Fig. 1 – Application structure

Приложение состоит из пяти функциональных модулей:

- модуль температурного контроля: обеспечивает мониторинг и визуализацию температурных параметров в четырех зонах (бункер и три зоны экструдера);

- модуль ведения журнала: фиксирует все значимые события технологического процесса с временной меткой;

- модуль управления подачей гранул: предусматривает ручной и автоматический режимы подачи гранул в промежуточный бункер;

- модуль управления шнеком: обеспечивает задание параметров вращения и контроль состояния экструзионного узла;

- модуль видеочамеры: интегрирует средства захвата и отображения видеопотока с USB-камеры. В текущей версии реализована базовая визуализация, однако архитектура предусматривает дальнейшее расширение функционала: анализ качества поверхности напечатанного изделия, обнаружение дефектов (прогиб, отслоение, перегрев) и синхронизация видео с данными температуры и подачи для пост-анализа. Такой подход соответствует современным тенденциям в аддитивном производстве, где визуальный

мониторинг становится ключевым элементом системы контроля качества.

Представленная архитектура программного обеспечения фокусируется на структурных и организационных аспектах системы, что соответствует общепринятой практике в публикациях по проектированию программных систем [11]. Такой подход позволяет сосредоточиться на ключевых принципах модульности и взаимодействия компонентов без отвлечения на детали пользовательского интерфейса. Более детальное рассмотрение визуализации процессов и элементов управления будет представлено в последующей публикации, посвященной реализации программного обеспечения.

Схема взаимодействия модулей (рис. 2) демонстрирует, что связь между ними организована преимущественно через главный класс-координатор (App) и механизм вызова методов. Например, любой модуль для записи события вызывает метод логирования, предоставляемый координатором. Такая организация предотвращает прямое взаимодействие модулей друг с другом, реализуя принцип слабого сцепления. Для обмена данными о состоянии системы (например, о текущем объеме гранул) используется общий контекст, управляемый главным классом.



Рис. 2 – Схема взаимодействия программных модулей

Fig. 2 – The scheme of interaction of software modules

Для реализации динамического поведения системы (плавное изменение температуры, периодическая подача материала) используется событийная модель с периодическим обновлением состояния. Такой подход гарантирует выполнение всей логики в едином потоке выполнения, что упрощает разработку и исключает необходимость в сложной синхронизации. Например, плавное изменение температуры до заданного значения, а процессы подачи гранул и экструзии моделируются как периодическое изменение объема материала в бункере.

Управление состоянием приложения реализовано с использованием объектно-ориентированных принципов: состояние аппаратного компонента (объем гранул) инкапсулировано в соответствующем модуле. Для сохранения этого состояния между сеансами работы применяется механизм постоянного хранения настроек, что позволяет восстанавливать текущий объем гранул при повторном запуске приложения. Это важная функция для имитации непрерывного производственного цикла.

Архитектура графического интерфейса построена на принципах, близких к паттерну «Модель-Представление-Контроллер» (MVC), в его упрощенной форме, характерной для десктопных приложений. «Представлением» являются визуальные компоненты: JLabel, JSpinner, JProgressBar, а также пользовательские JPanel для отображения графиков и видеопотока, в то время как «Моделью» выступают данные приложения, такие как текущая и целевая температуры, объем гранул в бункере, параметры шнека. Логика «Контроллера» реализована через обработчики событий (например, ChangeListener у JSpinner или ActionListener у кнопок), которые изменяют состояние модели и инициируют обновление представления.

Для строгого описания глобальной архитектуры системы введена ее формальная модель, инвариантная к языку реализации. Система представляется как кортеж (1):

$$S = (C, I, M, c_0, \Gamma), \quad (1)$$

где

- $C = \{c_0, c_1, c_2, c_3, c_4, c_5\}$ – множество компонентов системы. В данной реализации c_0 выступает в роли центрального координатора системы. Остальные компоненты соответствуют пяти функциональным модулям: управление температурами, логирование, управление подачей гранул, управление шнеком и видеомониторинг;

- $I(c_i)$ – интерфейс компонента c_i , определяемый его публичными методами. Для координатора c_0 ключевым методом является сервис логирования;

- $M = \{m_{evt}, m_{cmd}\}$ – множество типов сообщений (события и команды изменения состояния);

- Γ – правило взаимодействия, запрещающее прямую коммуникацию между любыми двумя компонентами $\forall i, j > 0, i \neq j$. Все взаимодействия должны проходить через координатор c_0 .

Архитектура реализует топологию «звезда», где координатор c_0 выполняет две ключевые роли:

1) фабрика и владелец компонентов – все модули создаются в конструкторе App() и хранятся как его приватные поля;

2) единая точка входа – модули взаимодействуют с системой только через вызовы сервисов координатора (например, App.this.logEvent(...)).

Следует отметить, что демонстрационный прототип, реализующий предложенную архитектуру, выполнен в имитационном режиме без подключения к реальному оборудованию. Все процессы (нагрев, подача, экструзия) моделируются программно. Это ограничение прототипа, однако сама архитектура предусматривает возможность интеграции с физическими датчиками и исполнительными механизмами через промышленные протоколы (Modbus, CAN и др.).

Таким образом, разработанная модульная архитектура обеспечивает гибкость, расширяемость и удобство управления, что делает ее перспективной основой для создания полноценного автоматизированного рабочего места оператора FGF-печати. Научная новизна работы заключается в предложении кроссплатформенной программной архитектуры управления шнековым экструдером, построенной на принципах слабого зацепления, централизованной

координации модулей и четкого разделения ответственности. Практическая значимость подтверждается возможностью ее адаптации под различные исследовательские и опытно-промышленные задачи. В перспективе планируется интеграция данной архитектуры с реальным оборудованием и расширение функционала за счет добавления сценарного режима управления и поддержки промышленных коммуникационных протоколов.

Литература

1. Л.В. Беляев, А.В. Аборкин, Введение в аддитивные технологии. Владимирский государственный университет имени А.Г. и Н.Г. Столетовых, Владимир, 2023. 248 с.
2. A. Gebhardt, J. Hötter, Additive Manufacturing: 3D Printing for Prototyping and Manufacturing. Carl Hanser Verlag GmbH & Company KG, Munich, 2016. 611 p.
3. Особенности «Шнековой» FGF печати. 3Dtoday, 2024. – URL: <https://3dtoday.ru/blogs/hero124/osobennosti-snekovoi-fgf-pecati-cast-1> (дата обращения: 22.12.2025).
4. C. Rackwitz, S. Scholz, M. Vyhnał, P. Hirsch, R. Schlimper, M. Zscheyge, Prog. Addit. Manuf., 9, 4, 665–674 (2024). DOI: 10.1007/s40964-024-00659-0.
5. F. Bandinelli, E. Tito, E. Parisi, L. Peroni, M. Scapin, Compos. Part B Eng., 294, Article 112147 (2025). DOI: 10.1016/j.compositesb.2025.112147.
6. Ю.И. Литвинец, Технологические и энергетические расчеты при переработке полимеров экструзией. Уральский государственный лесотехнический университет, Екатеринбург, 2010. 56 с.
7. A. Curmi, A. Rochman, Polymer, 330, Article 128459 (2025). DOI: 10.1016/j.polymer.2025.128459.
8. Э. Брауде, Технология разработки программного обеспечения. Питер, Санкт-Петербург, 2004. 655 с.
9. Ф.А. Данилкин, А. В. Новиков, Известия ТулГУ. Технические науки, 9, 2, 79-89 (2013).
10. N. Munir, T. de Lima, M. Nugent, M. McAfee, Functional Composite Materials, 5, Article 14 (2024). DOI: 10.1186/s42252-024-00063-5.
11. А.М. Пищухин, М.Н. Шевченко, Вестник СамГТУ. Серия: Технические науки, 2, 128-134 (2010).
12. F. Schade, M. Kreutzer, E. Mühlbeier, E. Gerlitz, P. Gönnheimer, J. Fleischer, J. Becker, Procedia CIRP, 120, 601-606 (2023). DOI: 10.1016/j.procir.2023.09.045.
13. B. Vogel-Heuser, J. Fischer, S. Feldmann, S. Ulewicz, S. Rösch, Journal of Systems and Software, 131, 35-62 (2017). DOI: 10.1016/j.jss.2017.05.051.
14. А.В. Левушкин, М.К. Турчанинов, А.А. Жиганов, В.В. Ермолаева, Молодой ученый, 25, 96-98 (2018).
15. Д.Р. Чиганов, Вестник науки, 4, 7, 276-278 (2023).

References

1. L.V. Belyaev, A.V. Aborkin, Introduction to additive technologies. Vladimir State University named after A.G. and N.G. Stoletov, Vladimir, 2023. 248 p.
2. A. Gebhardt, J. Hötter, Additive Manufacturing: 3D Printing for Prototyping and Manufacturing. Carl Hanser Verlag GmbH & Company KG, Munich, 2016. 611 p.
3. Features of "Screw" FGF printing. 3Dtoday, 2024. – URL: <https://3dtoday.ru/blogs/hero124/osobennosti-snekovoi-fgf-pecati-cast-1> (date of request: 12.22.2025).
4. C. Rackwitz, S. Scholz, M. Vyhnał, P. Hirsch, R. Schlimper, M. Zscheyge, Prog. Addit. Manuf., 9, 4, 665–674 (2024). DOI:10.1007/s40964-024-00659-0.
5. F. Bandinelli, E. Tito, E. Parisi, L. Peroni, M. Scapin, Compos. Part B Eng., 294, Article 112147 (2025). DOI: 10.1016/j.compositesb.2025.112147.
6. Y.I. Litvinets, Technological and energy calculations in polymer extrusion processing. Ural State Forestry Engineering University, Yekaterinburg, 2010. 56 p.
7. A. Curmi, A. Rochman, Polymer, 330, Article 128459 (2025). DOI: 10.1016/j.polymer.2025.128459.
8. E. Braude, Software Development technology. Piter, St. Petersburg, 2004. 655 p.
9. F.A. Danilkin, A.V. Novikov, Izv. TulGU. Technical Sciences, 9, 2, 79-89 (2013).
10. N. Munir, T. de Lima, M. Nugent, M. McAfee, Functional Composite Materials, 5, Article 14 (2024). DOI: 10.1186/s42252-024-00063-5.
11. A.M. Pischukhin, M.N. Shevchenko, Vestn. SamGTU. Series: Technical Sciences, 2, 128-134 (2010).
12. F. Schade, M. Kreutzer, E. Mühlbeier, E. Gerlitz, P. Gönnheimer, J. Fleischer, J. Becker, Procedia CIRP, 120, 601-606 (2023). DOI: 10.1016/j.procir.2023.09.045.
13. B. Vogel-Heuser, J. Fischer, S. Feldmann, S. Ulewicz, S. Rösch, Journal of Systems and Software, 131, 35-62 (2017). DOI: 10.1016/j.jss.2017.05.051.
14. A.V. Levushkin, M.K. Turchaninov, A.A. Zhiganov, V.V. Ermolaeva, Young Scientist, 25, 96-98 (2018).
15. D.R. Chiganov, Bulletin of Science, 4, 7, 276-278 (2023).

© **А. О. Фирсин** – аспирант кафедры Технологии металлов и авиационного материаловедения (ТМиАМ), Самарский национальный исследовательский университет имени академика С. П. Королева (Самарский университет), Самара, Россия, firsin.ao@ssau.ru; **Е. А. Носова** – д.т.н., заведующий кафедрой ТМиАМ, Самарский университет, nosova.ea@ssau.ru.

© **А. О. Firsin** – PhD-student, Department of Metal Technology and Aviation Materials Science (MTAMS), Samara National Research University named after Academician S. P. Korolev (Samara University), Samara, Russia, firsin.ao@ssau.ru; **E. A. Nosova** – Doctor of Sciences (Technical Sci.), Head of the MTAMS Department, Samara University, nosova.ea@ssau.ru.

Дата поступления рукописи в редакцию – 14.01.26.

Дата принятия рукописи в печать – 08.04.26.