

УДК: 004.89

**Худайберидева Г. Б., магистр, ассистент кафедры
«Информатика и информационные технологии»
Московский Политехнический Университет,
Россия, г. Москва**

**Кожухов Д. А., магистр, ассистент кафедры
«Информатика и информационные технологии»
Московский Политехнический Университет,
Россия, г. Москва**

**Пименкова А. А., студент-бакалавр кафедры
«Информатика и информационные технологии»
Московский Политехнический Университет,
Россия, г. Москва**

МИКРО-LLM КАК АССЕМБЛЕР ДЛЯ ПРОМЫШЛЕННОЙ АВТОМАТИКИ: ГЕНЕРАЦИЯ НИЗКОУРОВНЕВОГО КОДА ПЛК/SCADA ИЗ ЕСТЕСТВЕННОГО ЯЗЫКА НА МАЛОМОЩНЫХ УСТРОЙСТВАХ

Аннотация: Предлагается инновационный подход к применению микроскопических языковых моделей (микро-LLM) в промышленной автоматизации. Вместо традиционной генерации текста, микро-LLM позиционируются как специализированные ассемблеры для трансляции коротких команд оператора на естественном языке напрямую в исполняемый низкоуровневый код контроллеров (ПЛК) и систем диспетчеризации (SCADA). Ключевыми аспектами являются ультраограниченный словарь, гарантированная детерминированность вывода и глубокая интеграция с runtime-средой промышленных систем. Фокус на эффективность исполнения на ресурсоограниченных устройствах и повышение интуитивности человеко-машинного интерфейса (HMI) без ущерба надежности. Рассматриваются

архитектурные принципы, требования к безопасности и потенциальное влияние на операторские интерфейсы АСУ ТП.

Ключевые слова: микро-LLM; промышленная автоматизация; ПЛК; SCADA; генерация кода; естественный язык; человеко-машинный интерфейс (HMI); детерминированность; ресурсограниченные устройства; низкоуровневое программирование; ST; FBD; Ladder Diagram.

Khudaiberideva G. B.

**master and department assistant at the department of
"Computer Science and Information Technology"
Moscow Polytechnic University
Moscow, Russia**

Kozhukhov D. A.

**master and department assistant at the department of
"Computer Science and Information Technology"
Moscow Polytechnic University
Moscow, Russia**

Pimenkova A. A.

**bachelor's student at the department of
"Computer Science and Information Technology"
Moscow Polytechnic University
Moscow, Russia**

**MICRO-LLM AS AN ASSEMBLER FOR INDUSTRIAL AUTOMATION:
LOW-LEVEL PLC/SCADA CODE GENERATION FROM NATURAL
LANGUAGE ON LOW-POWER DEVICES**

Annotation: An innovative approach to the application of microscopic language models (micro-LLM) in industrial automation is proposed. Instead of traditional text generation, micro-LLMs are positioned as specialized assemblers for translating short

operator commands in natural language directly into executable low-level code for controllers (PLCs) and dispatch systems (SCADA). The key aspects are an ultra-limited vocabulary, guaranteed deterministic output, and deep integration with the runtime environment of industrial systems. Focus on performance efficiency on resource-limited devices and enhance the intuitiveness of the human-machine interface (HMI) without compromising reliability. Architectural principles, security requirements, and potential impact on automated process control system operator interfaces are considered.

Keywords: *micro-LLM; industrial automation; PLC; SCADA; code generation; natural language; Human-machine interface (HMI); determinism; resource-limited devices; low-level programming; ST; FBD; Ladder Diagram.*

Введение

Человеко-машинные интерфейсы (HMI) в системах автоматизации технологических процессов (АСУ ТП) традиционно требуют от операторов знания специфических языков программирования контроллеров (ПЛК), таких как Ladder Diagram (LD), Function Block Diagram (FBD), Structured Text (ST), или сложных систем команд SCADA [1]. Это создает барьер для эффективного взаимодействия и повышает риск ошибок при выполнении нестандартных операций или тонкой настройке параметров. Параллельно наблюдается рост возможностей языковых моделей (LLM), однако их применение в критически важных промышленных средах ограничивается вычислительной сложностью, стохастичностью вывода и отсутствием гарантированной детерминированности [2]. Возникает потребность в принципиально ином подходе, сочетающем интуитивность естественного языка (ЕЯ) с надежностью и предсказуемостью исполнения на уровне промышленных контроллеров.

Концепция микро-LLM как специализированного транслятора

Инновация предлагаемого подхода заключается в кардинальном сдвиге парадигмы применения LLM. Вместо универсальных моделей, генерирующих текстовые контент, предлагается использовать микро-LLM – крайне малые по размеру и узкоспециализированные модели, чья

единственная функция заключается в трансляции коротких, структурированных команд оператора на ЕЯ напрямую в валидный, исполняемый низкоуровневый код целевой промышленной платформы [3]. Аналогия с ассемблером здесь уместна: подобно тому, как ассемблер транслирует мнемонические коды в машинные инструкции, микро-LLM выступает в роли транслятора между ограниченным подмножеством ЕЯ и конкретными инструкциями ПЛК (например, команда SET_VALVE V101 OPEN) или вызовами функций SCADA API (например, `scada.setpoint("Reactor_Temp", 150.0)`) [4]. Критически важным отличием от генеративных LLM является полный отказ от креативности или интерпретации в пользу строгой, предопределенной трансляции.

Фундаментальные требования к архитектуре микро-LLM-транслятора

Успешная реализация концепции предъявляет жесткие требования к архитектуре микро-LLM.

Ультра-специализированный словарь и онтология. Словарь модели должен быть строго ограничен терминологией конкретного технологического процесса, установки или даже отдельного контура управления [5]. Это включает: номенклатуру технологических переменных (температура, давление, расход); идентификаторы оборудования (насос P101, клапан V203); допустимые действия (увеличить, уменьшить, открыть, закрыть, установить); единицы измерения и диапазоны значений; допустимые модификаторы (медленно, до достижения). Семантическая онтология, заложенная в модель, должна однозначно отображать термины ЕЯ на конкретные программные сущности в среде ПЛК/SCADA [6].

Гарантированная детерминированность вывода. Стохастическая природа генеративных LLM неприемлема в промышленной автоматизации [7, с. 201]. Вывод микро-LLM должен быть строго детерминированным: одна и та же входная команда ЕЯ при идентичных условиях всегда

порождает один и тот же выходной код ПЛК/SCADA. Это достигается через: архитектурный выбор (например, трансформеры с детерминированным beam search ширины 1 или даже детерминированные конечные автоматы, обученные как нейросети); отказ от вероятностных методов генерации на выходном слое; строгую валидацию выходного кода на соответствие заранее определенным шаблонам и грамматикам целевых языков (ST, FBD командные последовательности) [8].

Интеграция с промышленным runtime. Микро-LLM не функционирует изолированно. Ее вывод должен напрямую интегрироваться в runtime-среду системы управления. Это подразумевает: наличие безопасного интерфейса для передачи сгенерированного кода в исполняющую среду ПЛК или ядро SCADA; механизмы верификации синтаксиса и семантики генерируемого кода перед исполнением; возможность работы в режиме реального времени с предсказуемым временем отклика; поддержку контекста текущего состояния системы (например, доступность только актуальных имен тегов и их типов данных) [9]. Интеграция может быть реализована как встраиваемый модуль в HMI-станцию или непосредственно в firmware ПЛК при достаточных ресурсах.

Оптимизация для маломощных устройств. Ключевое преимущество микро-LLM – способность функционировать на граничных устройствах (edge devices) или даже самих ПЛК, обладающих ограниченными вычислительными ресурсами (CPU, RAM) и энергопотреблением [10]. Это требует экстремальной оптимизации модели: минимального размера (порядка мегабайт или даже килобайт); использования специализированных форматов данных и операторов (INT8 квантование, sparse-матрицы); аппаратного ускорения при наличии (например, NPU в современных ПЛК); эффективных алгоритмов вывода без итеративных дорогостоящих вычислений [11].

Потенциальное воздействие на человеко-машинные интерфейсы (HMI)

Внедрение микро-LLM-трансляторов способно революционизировать HMI в АСУ ТП. Оператор получает возможность формулировать команды на естественном языке ("поднять температуру в реакторе R-101 до 85 °С с градиентом 2 °С/мин"), минуя сложные мнемосхемы или экраны программирования ПЛК [12]. Это снижает когнитивную нагрузку; ускоряет выполнение нестандартных операций; потенциально уменьшает количество ошибок, вызванных неверной интерпретацией сложных интерфейсов. Однако, это не подразумевает замену традиционных HMI, а скорее их дополнение мощным инструментом для адресных управляющих воздействий и параметрической настройки [13]. Безопасность остается приоритетом: команды ЕЯ должны вводиться в строго контролируемом контексте с обязательной визуализацией генерируемого кода и запросом подтверждения перед исполнением критических операций.

Вопросы безопасности и надежности

Применение нейросетевых моделей в критических системах требует особого внимания к безопасности и надежности.

Защита от недопустимых команд. Микро-LLM должна обладать встроенными механизмами отвержения команд, выходящих за пределы ее словаря или онтологии; команд, нарушающих заданные технологические ограничения или правила безопасности; семантически неоднозначных или неполных команд [14]. Реакцией на такую команду должен быть четкий отказ с указанием причины, а не попытка генерации потенциально опасного кода.

Предсказуемость и верифицируемость. Детерминированность вывода является необходимым, но недостаточным условием. Сам процесс трансляции и генерируемый код должны поддаваться анализу и

верификации. Это включает: возможность логирования соответствия входной команды ЕЯ и выходного кода; использование статического анализа генерируемого кода на предмет промышленных стандартов безопасности (например, IEC 61131-3, ISA-88); строгую изоляцию процесса трансляции от основной системы управления [15].

Управление доступом и аудит. Интерфейс ввода команд ЕЯ должен быть интегрирован в общую систему разграничения доступа АСУ ТП. Каждая команда, ее исходный текст на ЕЯ, сгенерированный код и факт исполнения должны регистрироваться в системном журнале для последующего аудита [16]. Привилегия на использование расширенных команд должна предоставляться дифференцированно.

Заключение

Предложенная концепция применения микро-LLM в качестве специализированных трансляторов естественного языка в низкоуровневый код ПЛК/SCADA представляет собой перспективное направление для эволюции человеко-машинных интерфейсов в промышленной автоматизации. Сдвиг фокуса с генерации текста на детерминированную трансляцию команд позволяет преодолеть ключевые ограничения генеративных LLM применительно к требованиям АСУ ТП: надежность, безопасность, предсказуемость и работа на ресурсоограниченных устройствах. Ключевыми факторами успешной реализации являются: разработка ультра-специализированных моделей с жестко ограниченным словарем; обеспечение абсолютной детерминированности вывода; глубокая и безопасная интеграция с runtime-средой промышленных контроллеров и SCADA-систем; строгие механизмы защиты от недопустимых команд и всеобъемлющий аудит. Внедрение подобных систем потенциально способно значительно повысить интуитивность и эффективность взаимодействия оператора с системой управления технологическим процессом, снизив барьер для выполнения точных

настройки и оперативных вмешательств, при строгом соблюдении принципов промышленной безопасности.

СПИСОК ЛИТЕРАТУРЫ:

1. Болтон У. Программируемые логические контроллеры / Пер. с англ. 6-е изд. М.: БИНОМ. Лаборатория знаний, 2018. 422 с.
2. Amodei D. et al. Concrete Problems in AI Safety // arXiv preprint arXiv:1606.06565. 2016.
3. Han S. et al. MicroNets: Neural Network Architectures for TinyML // Proceedings of the 4th MLSys Conference. 2021. P. 1-15.
4. Патрушев Л.И., Щербаков Г.Л. SCADA-системы: Учеб. пособие. СПб.: Лань, 2020. 320 с.
5. Warden P. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition // arXiv preprint arXiv:1804.03209. 2018.
6. Gruber T.R. A translation approach to portable ontology specifications // Knowledge acquisition. 1993. Vol. 5, № 2. P. 199-220.
7. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб.: БХВ-Петербург, 2015. 736 с.
8. Vaswani A. et al. Attention is All You Need // Advances in Neural Information Processing Systems 30 (NIPS 2017). 2017.
9. Поляков К.Ю. Программирование промышленных контроллеров: учеб. пособие. М.: Академия, 2019. 256 с.
10. Lin J. et al. MCUNet: Tiny Deep Learning on IoT Devices // Advances in Neural Information Processing Systems 33 (NeurIPS 2020). 2020.
11. Шумаков А.А. Промышленные сети и интерфейсы. Учебное пособие. Томск: Томский политехнический университет, 2017. 180 с.
12. Sheridan T.B. Human–Robot Interaction: Status and Challenges // Human factors. 2016. Vol. 58, № 4. P. 525-532.
13. Норенков И.П. Основы автоматизированного проектирования: учебник для вузов. 4-е изд. М.: Изд-во МГТУ им. Н.Э. Баумана, 2013. 430 с.

14. Романов В.П. Безопасность систем управления технологическими процессами. М.: Горячая линия – Телеком, 2018. 304 с.
15. IEC 61131-3:2013. Programmable controllers – Part 3: Programming languages. 3rd ed. Geneva: IEC, 2013.
16. Смирнов Н.Н. Информационная безопасность АСУ ТП: практические аспекты. М.: ДМК Пресс, 2021. 288 с.