

М. П. Романов, д-р техн. наук, директор Института кибернетики, m_romanov@mirea.ru,

А. М. Романов, канд. техн. наук, ассистент, romanov@mirea.ru,

Е. Н. Каширская, канд. техн. наук, доц., liza.kashirskaya@gmail.com,

В. А. Холопов, канд. техн. наук, доц., holopov@gmail.com, **А. И. Харченко**, аспирант,

Московский технологический университет

Новая архитектура систем управления исполнительного уровня для дискретного машиностроительного производства¹

Предложена оригинальная архитектура системы управления исполнительного уровня для дискретного машиностроительного производства, которая сочетает в себе использование SCADA-системы и системы исполнения производственных заданий, основанной на применении виртуальных Forth-машин. Предлагаемая архитектура позволяет объединять в один производственный процесс операции, обладающие разной степенью автоматизации, и проводить замену ручных операций автоматизированными без изменения технологической документации.

Ключевые слова: ERP, MES, SCADA, Forth, мониторинг оборудования, производственное задание, дискретное машиностроительное производство

Введение

В основе развития современного промышленного производства лежит требование повышения его гибкости и эффективности за счет внедрения информационных технологий [1]. Важный шаг в этом направлении — появление систем управления производственными процессами (MES), которые осуществляют оперативное планирование и документальное сопровождение всех этапов производства — от поставки со склада исходных материалов до отгрузки готовой продукции [2]. В иерархии средств автоматизации производства MES-системы занимают промежуточное положение между системами ресурсного планирования ERP и системами исполнительного уровня, такими как системы диспетчеризации и сбора данных SCADA и системы распределенного числового управления DNC (рис. 1) [3].

Связь MES- и SCADA-систем в непрерывных производствах на сегодняшний день вполне отработана и широко применяется во всем мире. Для дискретных машиностроительных производств такая универсальная технология пока отсутствует. Это связано с широкой номенклатурой выпускаемой продукции, высокой вариативностью применяемых дополнительных приспособлений, оснастки и управляющих программ. При этом, в отличие от непрерывных производств, дискретные машиностроительные производства редко бывают полно-

стью автоматизированными, а степень автоматизации отдельных операций может существенно варьироваться даже в рамках производственного процесса обработки деталей одного типа.

Сложившаяся ситуация приводит к тому, что подавляющее большинство имеющихся MES-систем предполагает лишь частичную автоматизацию своего взаимодействия с технологическим оборудованием. Такая частичная автоматизация, как правило, ограничивается загрузкой управляющих программ в соответствующие системы ЧПУ по команде оператора станка. Такая ситуация тормозит как повышение эффективности существующих, так и создание новых предприятий, обладающих существенно более высоким уровнем автоматизации.



Рис. 1. Иерархия средств автоматизации производства

¹ Данная работа выполнена в рамках прикладных научных исследований и экспериментальных разработок (ПНИЭР) по заказу Министерства образования и науки Российской Федерации (уникальный идентификатор ПНИЭР RFMEFI58016X0008).

Типы датчиков, используемых для мониторинга состояния оборудования на машиностроительном производстве, и их интерфейсы

Метод мониторинга	Требуемые датчики	Интерфейсы подключения датчиков
Контроль положения подвижных элементов высокотехнологического оборудования	Концевые датчики Датчики абсолютного положения Системы технического зрения Лазерные дальномеры	Дискретные входы 24В Цифровые интерфейсы SSI, BISS, EnDat Цифровой интерфейс Ethernet, полевые шины EtherCAT, Ethernet POWERLINK и др. Цифровые интерфейсы SSI, RS232, RS485. Аналоговый интерфейс 4...20 mA
Контроль скорости подвижных частей станков	Датчики скорости	Цифровой интерфейс EnDat, дискретные входы 24В, дифференциальные дискретные входы по стандарту RS422
Контроль усилий	Силовоймоментные датчики	Аналоговые интерфейсы 0...10В, 4...20 mA
Вибродиагностика	Акселерометры	Аналоговые интерфейсы 0...10В, 4...20 mA. Специальные модули обработки аналоговых сигналов, подключаемые к полевым шинам Modbus TCP, EtherCAT, Ethernet POWERLINK и др.
Контроль качества питающих напряжений	Реле контроля фаз Вольтметры Амперметры Комплексные модули контроля качества питающих напряжений	Дискретные входы 24В Аналоговые интерфейсы $\pm 10В$, 4...20 mA. Специальные модули, подключаемые к полевым шинам Modbus TCP, EtherCAT, Ethernet POWERLINK Аналоговые интерфейсы $\pm 10В$, 4...20 mA. Специальные модули, подключаемые к полевым шинам Modbus TCP, EtherCAT, Ethernet POWERLINK и др. Специальные модули, подключаемые к полевым шинам Modbus TCP, EtherCAT, Ethernet POWERLINK и др.
Контроль температуры	Датчики температуры Тепловизоры	Аналоговые интерфейсы $\pm 10В$, 4...20 mA, pt100, pt1000, входы термпар Аналоговый видеосигнал, цифровой интерфейс Ethernet
Контроль характеристик технических жидкостей и газов	Расходомеры Датчики давления Датчики уровня	Аналоговые интерфейсы $\pm 10В$, 4...20 mA, HART Аналоговые интерфейсы $\pm 10В$, 4...20 mA, HART Дискретные входы 24В Аналоговые интерфейсы $\pm 10В$, 4...20 mA
Контроль качества воздуха на рабочем месте	Датчики задымления Газоанализаторы Датчики влажности Датчики концентрации пыли	Дискретные входы 24В Аналоговые интерфейсы $\pm 10В$, 4...20 mA Аналоговые интерфейсы $\pm 10В$, 4...20 mA Цифровые интерфейсы USB, RS232, RS485, Ethernet. Аналоговый интерфейс 4...20 mA
Датчики контроля доступа персонала в опасную зону	Датчики открытия двери Световые барьеры Датчики присутствия/движения Лазерные сканеры (Лидары) Считыватели RFID-меток и магнитных карт	Дискретные входы 24В Дискретные входы 24В Дискретные входы 24В Цифровой интерфейс Ethernet, полевые шины Modbus TCP, EtherCAT и др. Цифровые интерфейсы Ethernet, RS232, RS485

Целью данного исследования является разработка универсальной архитектуры аппаратно-программных средств исполнительного уровня для систем управления дискретным машиностроительным производством. Такая архитектура позволила бы объединить на базе единой платформы все средства супервизорного мониторинга и управления станочным и робототехническим оборудованием, а также обеспечила бы единый интерфейс взаимодействия с MES-системой для рабочих мест, обладающих различной степенью автоматизации.

Архитектура аппаратного обеспечения

Главным требованием, предъявляемым к новой архитектуре, является требование использования единых аппаратно-программных средств для комплексного решения задач управления и мониторинга состояния оборудования на дискретном машиностроительном производстве.

Обеспечение качественного мониторинга состояния оборудования сопряжено с необходимостью

анализа показаний большого числа разнородных датчиков (табл. 1).

Выбор интерфейса для подключения датчиков и производственного оборудования

Поддержка столь широкой номенклатуры интерфейсов на базе одного устройства сбора данных является задачей крайне трудоемкой и экономически нецелесообразной. Наилучшим решением, хорошо зарекомендовавшим себя на практике, является использование распределенных модулей входов/выходов на базе полевых шин [4]. Таким образом, задача выбора полевой шины, которая ляжет в основу разрабатываемой универсальной архитектуры, является актуальной.

В качестве критериев для оценки пригодности полевой шины были выбраны распространенность на рынке среди производителей модулей промышленных входов-выходов, соответствие быстродействия классу решаемых задач, пропускная способность, возможность передачи данных по стандарт-

ным протоколам на базе TCP/IP, а также наличие открытого стандарта и готовых решений для реализации ведущих и ведомых устройств. Рассмотрим каждый из указанных выше критериев в отдельности.

♦ *Занимаемая доля рынка*

Доля рынка, занимаемая той или иной полевой шиной, определяет ее распространенность среди поставщиков и потребителей компонентов, а также косвенно говорит о широте номенклатуры различных типов входов-выходов, которые смогут быть в дальнейшем подключены с ее использованием к исполнительному уровню системы управления производством. В данной работе была использована информация о доле рынка, занимаемой каждой из рассматриваемых полевых шин, на основании исследований IMS Research по состоянию на 2013 г.

♦ *Быстродействие*

Быстродействие полевой шины, как правило, определяется минимальным временем цикла обмена данными и джиттером, возникающим при синхронизации подключенных к шине устройств. Чем меньше эти два параметра, тем с большей частотой и точностью система мониторинга сможет фиксировать работу оборудования. Это, в свою очередь, позволит учитывать при анализе возможных неисправностей и более динамичные процессы. В работе [5] дается оценка постоянных времени процессов, протекающих в системах управления, в зависимости от области автоматизации (табл. 2). Стоит отметить, что для корректности последующего анализа время цикла полевой шины должно быть, как минимум, в два раза меньше, чем постоянная времени наблюдаемого процесса, а джиттер должен быть, как минимум, в два раза меньше используемого времени цикла. Анализ табл. 2, в которой представлены характеристики датчиков для мониторинга, показывает, что постоянные времени в процессах, анализируемых системой мониторинга на дискретном машиностроительном производстве, лежат в диапазоне от 1 мс до 1 с.

Таблица 2

Типы датчиков, используемых для мониторинга состояния оборудования на машиностроительном производстве, и их интерфейсы

Область автоматизации	Минимальные значения постоянных времени процессов	Максимальные значения постоянных времени процессов
Системы хранения	30 мс	15 с
Конвейерные системы и другие простые системы управления. (большинство систем управления)	1 мс	1 с
Станки и робототехнические системы	50 мкс	50 мс
Системы электропривода	1 мкс	1 мс

В робототехнических системах могут протекать отдельные процессы, имеющие постоянные времени менее 1 мс, но их анализ, как правило, проводится в самой системе управления роботом с использованием аппаратно-программных средств, обладающих соответствующим быстродействием. Таким образом, можно сделать вывод, что для класса задач, решаемых системой мониторинга, необходимо, чтобы полевая шина обеспечивала цикл в диапазоне от 500 мкс до 500 мс с джиттером не хуже 250 мкс.

♦ *Пропускная способность и передача данных по протоколам на базе TCP/IP*

Для упрощения аппаратного обеспечения предлагаемой архитектуры и использования единых аппаратных средств как для мониторинга, так и для управления технологическим оборудованием предлагается использовать полевые шины не только для подключения датчиков, но и для передачи управляющих программ на роботы и станки с ЧПУ. Основными интерфейсами подключения для систем ЧПУ на сегодняшний день являются RS232, RS485 и Ethernet. Последний приобретает все большую популярность при взаимодействии между системами управления разных уровней. При этом Ethernet с использованием специальных интерфейсных модулей позволяет инкапсулировать в себя более медленные протоколы RS232 и RS485. По этой причине при выборе полевой шины рассматривались только шины с пропускной способностью не менее 100 Мбит/с, при этом особое внимание уделялось возможности прозрачной передачи через них трафика протоколов на базе TCP/IP. Такие шины в подавляющем большинстве случаев используются в станках с ЧПУ, подключаемых по интерфейсу Ethernet.

♦ *Наличие открытых решений*

Последним критерием при выборе шины явилось наличие открытого стандарта и готовых открытых реализаций для ведущего и ведомых устройств. Данный критерий косвенно показывает доступность полевой шины для сторонних разработчиков, а также относительные трудозатраты на реализацию на ее базе новых устройств.

♦ *Сравнение наиболее распространенных полевых шин на базе Ethernet*

Сравнение полевых шин по указанным выше критериям проводилось на основе материалов работ [5–7]. Результаты данного сравнения приведены в табл. 3.

Полностью соответствуют требованиям по быстродействию только две из рассмотренных шин: Ethernet POWERLINK и EtherCAT. По остальным критериям эти шины также сопоставимы. Выгодным отличием Ethernet POWERLINK от EtherCAT является ее распространение под лицензией BSD, которая не накладывает никаких существенных ограничений на разработчиков, и наличие готовых

Результаты сравнения полевых шин с точки зрения применения аппаратно-программных средств исполнительного уровня для систем управления дискретным машиностроительным производством

Критерий	PROFINET	EtherNet/IP	Modbus TCP	EtherCAT	Ethernet POWERLINK
Занимаемая доля рынка	30 %	30 %	17 %	7 %	9 %
Наличие открытого стандарта и лицензионные ограничения	Открытый стандарт, спецификация по платной подписке	Открытый стандарт, спецификация по платной подписке	Открытый стандарт с открытой спецификацией	Открытый стандарт, спецификация по платной подписке, наличие сборов за создание ведомых устройств	Открытый стандарт с открытой спецификацией
Наличие открытых бесплатных решений для реализации ведущего и ведомого устройств	Нет	Есть только для ведомого устройства	Есть	Есть для ведущего устройства, но без открытой лицензии	Есть
Возможность использования совместно с полевой шиной стандартных сетевых протоколов, основанных на стеке TCP/IP	Есть	Есть	Есть	Есть	Есть, отдельно от данных, требующих реального времени
Соответствие цикла обмена сети классу задач, решаемых системой управления исполнительного уровня	Соответствует при управлении в системах с малой динамикой	Соответствует только с расширением CIPsync	Соответствует при управлении в системах с медленной динамикой	Соответствует	Соответствует

реализаций под различные платформы, начиная от встраиваемых компьютеров и заканчивая ПЛИС. Последнее существенно упрощает для разработчиков станков интеграцию встроенных систем мониторинга и контроля неисправностей с системами управления, построенными на предложенной в данной работе архитектуре. В связи с этим в качестве основного средства информационного обмена с датчиками, станками ЧПУ и робототехническими системами предлагается выбрать именно шину Ethernet POWERLINK.

Выбор аппаратных средств реализации исполнительного уровня системы управления

С точки зрения задач мониторинга оптимальной программной платформой, хорошо зарекомендовавшей себя на практике, является SCADA-система. Большинство из представленных SCADA-систем имеют трехуровневую архитектуру [8], показанную на рис. 2.

Нижний уровень составляют точки сбора данных и формирования управляющих воздействий.

В качестве среднего уровня выступает SCADA-сервер, который агрегирует данные со всех программируемых логических контроллеров (ПЛК), ведет журналы сигналов и аварийных ситуаций, анализирует возможные сбои и неисправности, а также осуществляет супервизорное управление отдельными ПЛК.

Верхний уровень составляют операторские станции и иные компьютерные системы, использующие результаты работы SCADA-системы и формирующие для нее задания, в том числе MES-систем.

В основном описанная архитектура применяется на непрерывных рецептурных производствах, однако с рядом уточнений она может быть применена и к дискретному машиностроительному производству.

Главным отличием дискретного машиностроительного производства от непрерывного рецептурного является высокая вариативность задач, выполняемых на одном и том же рабочем месте в течение одной смены. Рецепт, как правило, позволяет изменить отдельные параметры той или иной технологической операции, такие как температура или концентрация, в то время как на дискретном машиностроительном производстве меняется состав и порядок операций, выполняемых на автоматизированном рабочем месте. Это существенно усложняет централизованное управление всеми рабочими местами с помощью SCADA-сервера, и поэтому нами предлагается установка точек сбора данных и формирования управляющих воздействий на каждом автоматизированном рабочем месте, с последующим подключением к ним датчиков системы мониторинга станков с ЧПУ, а также робототехнических систем. Наиболее подходящая ре-

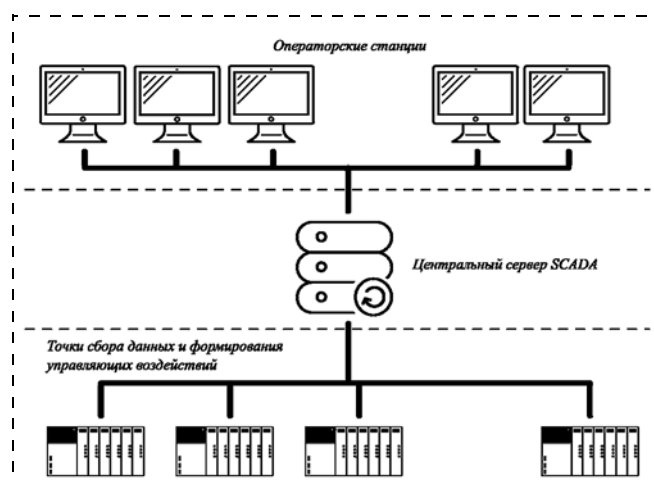


Рис. 2. Архитектура SCADA-систем

ализация точек сбора данных и формирования управляющих воздействий осуществляется на базе ПЛК или промышленных компьютеров, способных самостоятельно решать задачи управления оборудованием с ЧПУ и обеспечения правильного порядка действий при выполнении технологических операций. В последнем случае использование SCADA-сервера на среднем уровне оправдано только с точки зрения решения задач мониторинга, так как ПЛК исполнительного уровня могут взаимодействовать непосредственно с MES-системой, получая отсюда информацию о порядке действий и составе управляющих программ, которые необходимо выполнить в ходе той или иной производственной операции. Данный подход аналогичен по своей сути системам проектного менеджмента, применяемым для управления персоналом [9], с той лишь разницей, что здесь происходит взаимодействие не "человек—человек", а "машина—машина": MES ставит производственную задачу и устанавливает сроки ее выполнения, а промышленный контроллер подтверждает начало работы над задачей, контролирует заданный порядок действий и в конце сообщает об успешном выполнении задачи или о проблемах, приведших к невозможности ее завершения.

Связь между SCADA и ПЛК или промышленным компьютером наиболее целесообразно осуществлять по протоколу OPC UA [10], который на данный момент является основным стандартом для осуществления коммуникаций такого рода.

Вопрос выбора оптимального протокола связи между ПЛК и MES-системой требует дополнительных исследований. В данной работе для этих целей предлагается использовать HTTPS, так как он позволяет ПЛК взаимодействовать непосредственно с веб-интерфейсом пользователя MES-системы и

не требует для реализации этого взаимодействия внесения в MES-систему каких-либо изменений. Дополнительно к HTTPS предлагается использовать FTPS для передачи управляющих программ, так как этот файловый протокол реализован во многих современных ПЛК.

Таким образом, структурная схема аппаратных средств предлагаемой архитектуры исполнительного уровня системы управления дискретного машиностроительного производства будет иметь вид, представленный на рис. 3.

Архитектура программного обеспечения

Как отмечалось выше, для реализации функций мониторинга в рамках разработанной архитектуры предполагается использование SCADA-систем. Подробности применения этого подхода в данной работе рассматриваться не будут, так как он известен и широко освещен в литературе, например, в работах [8, 11].

Ключевой особенностью предлагаемой архитектуры с точки зрения программного обеспечения является прямое взаимодействие с MES-системой и использование специализированного языка программирования для описания технологических процессов, выполняемых на рабочем месте.

Выбор языка программирования для описания технологических процессов

Высокая вариативность технологических процессов на дискретном машиностроительном производстве не позволяет полностью автоматизировать весь технологический процесс [17]. На современных предприятиях, как правило, автоматизированы лишь основные технологические операции посредством систем ЧПУ, а вспомогательные технологические операции, связанные со сменой номенклатуры обрабатываемых деталей, автоматизируются крайне редко, что, в большинстве случаев, экономически не целесообразно. По этим причинам разработан язык описания основных и минимально необходимых вспомогательных операций, который реализуется в виде управляющих программ для систем ЧПУ, а для описания вспомогательных операций всего технологического процесса на сегодняшний день не разработано ни одного приемлемого языка. Применение для этих целей средств, предоставляемых SCADA-системами, представляется невозможным, так как они ориентированы на непрерывные рецептурные производства.

В качестве возможного решения в данной работе предлагается использование специального языка программирования, с помощью которого технолог сможет описать технологический процесс в целом, включая межоперационные вспомогательные операции, продиктованные необходимостью смены элемента номенклатуры выпускаемых изделий. Описание основных операций остается традици-

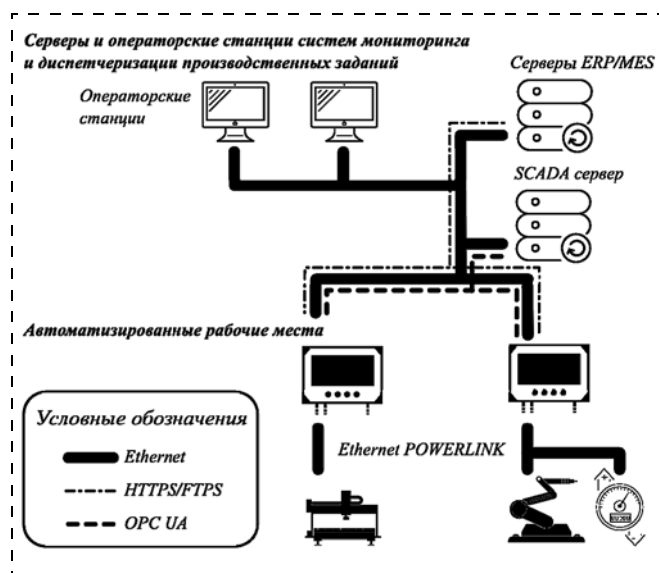


Рис. 3. Архитектура аппаратных средств исполнительного уровня системы управления дискретного машиностроительного производства

онным, и управляющие программы будут инкапсулированы как объекты в создаваемый язык программирования.

В процессе исполнения технологического процесса создаваемый язык программирования будет исполняться на ПЛК или на промышленном компьютере, установленном на автоматизированном рабочем месте. На этом языке будет определяться последовательность действий в ходе каждой производственной операции.

Чтобы обеспечить возможность применения данного подхода на частично автоматизированных рабочих местах, предлагается при необходимости выполнения той или иной ручной операции (например, замены приспособлений или установки заготовки) выводить оператору на экране соответствующую команду-запрос на выполнение нужной операции. Это позволит технологу использовать единые инструменты для описания всех производственных процессов вне зависимости от степени их автоматизации.

Чтобы выбрать язык программирования для описания технологических процессов, выполняемых на автоматизированных рабочих местах дискретного машиностроительного производства, сформируем предъявляемые к языку требования.

♦ *Задачная ориентированность*

Поскольку целевой аудиторией выбираемого языка с точки зрения создания программ является не программист, а технолог, то сам язык должен быть максимально задачно-ориентированным, что позволит переходить от описания одного производственного процесса к описанию какого-либо иного производственного процесса с использованием минимального количества дополнительного кода. Данное требование осложняется тем, что различные машиностроительные производства имеют множество разных специализаций, поэтому подбор одного задачно-ориентированного языка для всех возможных типов дискретных машиностроительных производств является труднорешаемой проблемой.

♦ *Возможность абстрагирования от конкретных аппаратных средств*

Замена и модернизация оборудования на предприятии является естественным процессом, который в большинстве случаев не должен приводить к существенной переработке технологических программ. Таким образом, на уровне выбираемого технологического языка должна существовать возможность описания технологической операции, с абстрагированием от конкретных моделей оборудования, установленного на автоматизированном рабочем месте, и от степени его автоматизации.

♦ *Стойкость к кибератакам*

Современные принципы построения информационных систем предприятия [1] требуют учета возможности кибератак и, следовательно, должны

предусматривать действия для их предотвращения на архитектурном уровне.

С позиции кибербезопасности использование внутреннего технологического языка требует гарантий невозможности его использования для создания троянских программ. Достичь этого можно, например, ограничив доступ к аппаратным средствам и возможность подключения внешних библиотек, используя изолированные виртуальные машины и т.д.

♦ *Требования к аппаратному обеспечению*

Поскольку целевыми платформами для выполнения технологических программ являются ПЛК и промышленные компьютеры, нельзя не учитывать ограничения их аппаратного обеспечения, такие как малый объем оперативной памяти, использование низкопроизводительных процессоров с пассивным охлаждением, малый объем энергонезависимой памяти и т.д. Например, именно высокие требования к аппаратным ресурсам являются наиболее существенным препятствием к применению для решения рассматриваемой задачи такого широко распространенного языка, как Java.

♦ *Перспективы использования Forth*

Проведенный сравнительный анализ языков программирования показал, что наиболее пригодным для рассматриваемой в статье задачи является язык Forth [12]. Этот язык программирования стэковых машин, предложенный Чарльзом Муром в 1971 г., не потерял своей актуальности до сих пор [13]. В основе Forth лежат понятия стэка как основного средства обмена данными в программном обеспечении и слова, которое можно описать как совокупность других ранее объявленных слов. Основная идеология Forth [12] заключается в том, чтобы на основе базовых слов, реализованных в системе на этапе конфигурации, создать задачно-ориентированный язык программирования, оптимально подходящий для требуемой предметной области. В связи с этим Forth практически идеально удовлетворяет первому из описанных выше критериев, позволяя, с одной стороны, использовать унифицированные программные средства для выполнения технологических программ, а с другой — создавать язык, наилучшим образом соответствующий специализации конкретного производства. Помимо этого, Forth позволяет сделать код максимально похожим на естественный язык [12], применяемый в технологических картах, что будет способствовать сокращению времени на переучивание сотрудников предприятий.

Абстрагирование от конкретных аппаратных средств в языке Forth достигается использованием различных базовых словарей для разных исполнительных устройств. При этом технолог, создающий итоговую программу, может пользоваться для тестирования базовым словарем обобщенной модели оборудования каждого типа, а в процессе выпол-

нения ПЛК или промышленный компьютер, установленный на рабочем месте, будет использовать свой словарь, в котором учтены особенности реализации каждого из слов на основе учета возможностей конкретного аппаратного обеспечения.

Для выполнения кода Forth используются специальные стэковые виртуальные машины, исполнительный код которых, как правило, напрямую транслируется в базовые слова. Таким образом, варьируя набор базовых слов, можно гибко ограничивать возможности влияния программы, написанной на Forth, на другие компоненты системы, обеспечивая тем самым требуемый уровень кибербезопасности.

Как было отмечено выше, Forth тривиальным образом транслируется в машинный код, что обеспечивает его высокое быстродействие. А стэковые машины, используемые для его исполнения, имеют наиболее простую среди распространенных виртуальных машин архитектуру, что позволяет запускать программы на Forth не только на промышленных компьютерах и ПЛК, но даже на бюджетных микроконтроллерах [14] и ПЛИС [15].

Пример технологической программы на Forth представлен в листинге 1.

Листинг 1. Пример технологической программы на Forth для фрезеровки цилиндра, оформленный в виде нового слова "ОтфрезероватьЦилиндр"

ОтфрезероватьЦилиндр:

```
"Загрузить деталь в станок" ЗапросОператору
Станок ["cyl.cnc" Запустить_CNC]
Станок ["G01 X = %Радиус, Y = %Высота" Вставить_
ПараметрыВСтроку КомандаCNC]
Робот ["get_result.cnc" КомандаCNC]
```

;

Как показывает листинг 1, Forth успешно позволяет совмещать в одном коде команды, выполняемые оператором, и управляющие программы, выполняемые на станке и погрузочном роботе.

Выбор программных средств реализации исполнительного уровня системы управления

Предлагаемая архитектура программного обеспечения исполнительного уровня системы управления дискретным машиностроительным производством представлена на рис. 4.

На верхнем уровне системы располагается серверное программное обеспечение (ПО) систем SCADA, MES и ERP. ПЛК или промышленный компьютер, установленный на каждом автоматизированном рабочем месте, получает из MES-системы по протоколу HTTPS данные о времени начала следующей операции и о ее параметрах. По протоколу FTP контроллер автоматизированного рабочего места загружает все необходимые для выполнения следующей операции управляющие программы для роботов и систем ЧПУ, а также технологическую программу операции на языке Forth. За обмен с MES и ERP отвечает блок обмена про-

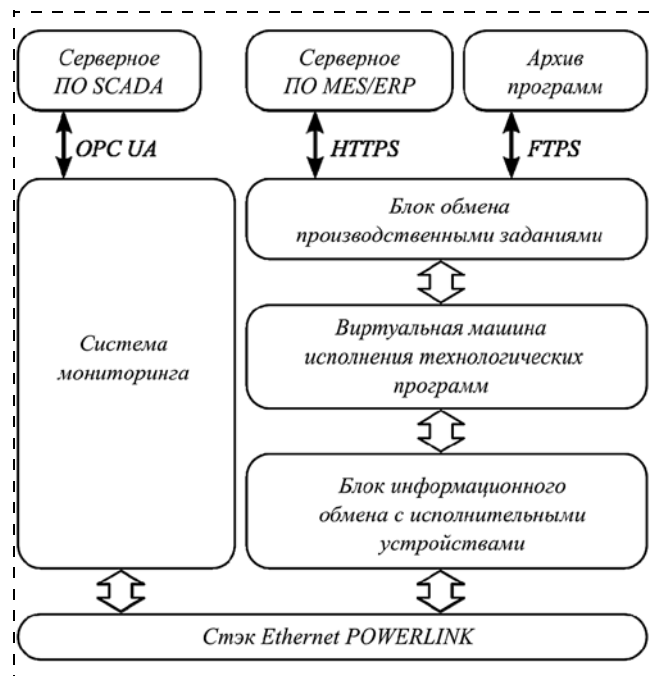


Рис. 4. Структурная схема программного обеспечения исполнительного уровня системы управления дискретным машиностроительным производством

изводственными заданиями. Как только наступает плановое время выполнения операции, технологическая программа поступает в виртуальную машину, которая ее исполняет, определяя последовательность действий в рамках одной операции и возможную вариативность исполнения операций.

К виртуальной машине подключен блок информационного обмена с исполнительными устройствами, который позволяет с помощью базовых слов Forth загружать управляющие программы в систему ЧПУ, управлять погрузочными или сборочными роботами, выдавать команды и запросы оператору через средства человеко-машинного интерфейса. По сути, блок информационного обмена с исполнительными устройствами реализует необходимый слой абстракции между технологическими программами и реальным оборудованием. По завершении технологической программы виртуальная машина делает отметку в MES-системе о завершении операции в блоке обмена производственными заданиями.

Параллельно и независимо от виртуальной машины исполнения технологических программ работает система мониторинга, которая собирает данные с датчиков, выставляет им метки времени и пересылает на сервер SCADA-системы по протоколу OPC UA.

Единственным элементом взаимодействия системы мониторинга с виртуальной машиной исполнения технологических операций является возможность последней сообщать об ошибках, возникающих при выполнении программ, в систему мониторинга для последующей фиксации этих данных в журнале аварийных сообщений.

Для проверки реализуемости предложенной архитектуры был поставлен эксперимент. На сервере с 64-битной архитектурой Intel в операционной системе Linux Ubuntu 14.04 LTS была развернута открытая система управления производством Odoo 9 Community, оснащенная возможностью выполнения отдельных функций MES-системы, таких как оперативное планирование и формирование производственных заданий.

В Odoo было создано производственное задание, включающее в себя три последовательные операции. В описании каждой операции была дана ссылка на производственную программу, расположенную на FTP-сервере.

На базе трех промышленных компьютеров В & R PP65 были реализованы точки сбора данных и формирования управляющих воздействий. Выбор контроллеров В & R был обоснован тем, что они оснащены встроенной системой ЧПУ, которая позволила провести эксперимент без использования дополнительного производственного оборудования. К системе ЧПУ были подключены по четыре исполнительных двигателя, с помощью которых обрабатывались управляющие программы. Тестирование программного кода промышленных компьютеров осуществлялось по методике, описанной в работе [16]. На рис. 5 представлено фото испытательного стенда.

В ходе эксперимента контроллеры поочередно выполняли производственные задачи, загрузив соответствующие управляющие программы в систему ЧПУ и сделав соответствующую отметку в системе Odoo после ее завершения.

Проведенный эксперимент показал реализуемость предложенного в данной работе подхода.

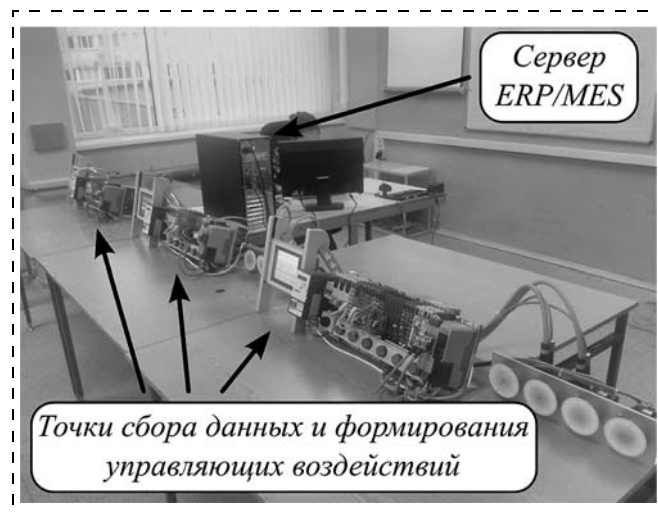


Рис. 5. Фото испытательного стенда

В данной работе предложена оригинальная архитектура реализации системы управления исполнительного уровня для дискретного машиностроительного производства. Она предполагает сочетание традиционного подхода, подразумевающего использование SCADA-системы для осуществления мониторинга состояния производственного оборудования, и системы исполнения производственных заданий, основанной на применении виртуальных Forth машин на каждом автоматизированном рабочем месте. Предлагаемая архитектура не только позволяет успешно объединять в единый производственный процесс операции, обладающие разной степенью автоматизации, но и обеспечивает возможность в ходе модернизации проводить замену ручных операций автоматизированными без изменения технологической документации. Проведенные экспериментальные исследования показывают реализуемость данного подхода и перспективность продолжения работ по данной тематике.

Список литературы

1. **Drath R., Horch A.** Industrie 4.0: Hit or hype? [industry forum] // IEEE industrial electronics magazine. 2014. Т. 8. № 2. С. 56—58.
2. **McClellan M.** Applying manufacturing execution systems. CRC Press, 1997.
3. **Sauter T.** The continuing evolution of integration in manufacturing automation // IEEE Industrial Electronics Magazine. 2007. Т. 1. № 1. С. 10—19.
4. **Faldella E., Tucci P.** A model-driven approach to automated diagnosis of industrial distributed I/O systems based on fieldbus technologies Information // Communication and Automation Technologies: XXII International Symposium on, ICAT (2009). Bosnia, 2009. С. 1—7.
5. **Lachello L.** et al. System Comparison // The 5 Major Technologies Industrial Ethernet Facts. 2013. Iss. 2.
6. **Industrial Automation Open Networking Alliance and Luder A.** IAONA Handbook — Industrial Ethernet: (with a List of Producers) // IAONA, 2005.
7. **Денисенко В.** Протоколы и сети Modbus и Modbus TCP // СТА. 2000. № 4. С. 90—94.
8. **Stouffer K., Falco J., Scarfone K.** Guide to industrial control systems (ICS) security // NIST special publication. 2011. Т. 800. № 82. С. 16.
9. **Mishra A., Mishra D.** Software project management tools: a brief comparative view // ACM SIGSOFT Software Engineering Notes. 2013. Т. 38. № 3. С. 1—4.
10. **Hannelius T., Salmenpera M., Kuikka S.** Roadmap to adopting OPC UA // 2008 6th IEEE International Conference on Industrial Informatics. Daejeon, 2008. С. 756—761.
11. **Boyer S. A.** Scada: Supervisory Control and Data Acquisition // International Society of Automation. USA, 2009.
12. **Brodie L.** Thinking Forth. Punchy Pub, 2004.
13. **Riedel S., Bošnjak M., Rocktschel T.** Programming with a Differentiable Forth Interpreter // arXiv preprint arXiv:1605.06640. 2016.
14. **De Ceballos F.** A Minimal Development Environment for the AVR Processor // 17th EuroForth Conference. Schloss Dagstuhl. 2001.
15. **Bowman J., Garage W.** J1: a small Forth CPU Core for FPGAs / EuroForth 2010 Conference Proceedings. 2010. С. 43—46.
16. **Басок Б. М., Захаров В. Н., Френкель С. Л.** Об оценке чувствительности к сбоям тестов программ // Российский технологический журнал. 2016. Т. 4, № 2 (11). С. 39—44.
17. **Холопов В. А., Голубцов И. Н.** Классификация автоматизированных производств для определения уровня и метода их автоматизации // Ползуновский вестник. Электронная версия. 2012. Т. 1. С. 315—317.

A Novel Architecture for the Executive-Level Control Systems for a Discrete Machinery Production

M. P. Romanov, m_romanov@mirea.ru✉, **A. M. Romanov**, romanov@mirea.ru,
E. N. Kashirskaya, liza.kashirskaya@gmail.com, **V. A. Kholopov**, holopov@gmail.com, **A. I. Kharchenko**,
Moscow Technological University, Institute of Cybernetics, Moscow, 119454, Russian Federation

Corresponding author: **Romanov Mikhail P.**, D. Sc.,
Director of the Institute of Cybernetics, Moscow Technological University,
Institute of Cybernetics, Moscow, 119454, Russian Federation, e-mail: m_romanov@mirea.ru

Received on September 09, 2016

Accepted on September 16, 2016

This paper presents a novel architecture of the executive-level control system for management of the discrete machinery productions. It combines the use of a supervisory control and a data acquisition system, and the tasks of the execution system based on the use of virtual Forth-machine. The widely used solutions for modern manufacturing are based on interaction of the manufacturing execution system directly with the supervisory control and data acquisition system, but this is not suitable for a discrete machinery manufacture, because of complexity of its production process. In the proposed architecture the supervisory control and data acquisition system are used for the online state monitoring of the CNC-machines hardware, signal logging and fault prediction. The task execution system downloads the manufacturing task, designed as the Forth program, from the manufacturing execution system and uploads it into the virtual machine, which controls the sequence of operations during one manufacturing task. Besides, the task execution system includes information exchange module, which ensures a software abstraction layer between the virtual machine and the hardware. The proposed architecture allows us to combine in one manufacturing process operations with a different degree of automation, and ensures replacement of the manual processes with automated ones without changing the manufacturing tasks documentation. The experimental results show that the proposed architecture is implementable, which determines the relevance of the further development in this direction.

Keywords: ERP, MES, SCADA, Forth, hardware monitoring, manufacturing task, discrete machinery manufacturing

Acknowledgements: This work was done as a part of the applied research and experimental development commissioned by the Ministry of Education and Science of the Russian Federation (the unique identifier is RFMEFI58016X0008).

For citation:

Romanov M. P., Romanov A. M., Kashirskaya E. N., Kholopov V. A., Kharchenko A. I. A Novel Architecture for the Executive-Level Control Systems for a Discrete Machinery Production, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2017, vol. 18, no. 1, pp. 64–72.

DOI: 10.17587/mau.18.64-72

References

1. **Drath R., Horch A.** Industrie 4.0: Hit or hype?, *IEEE industrial electronics magazine*, 2014, vol. 8, no. 2, pp. 56–58.
2. **McClellan M.** Applying manufacturing execution systems, CRC Press, 1997.
3. **Sauter T.** The continuing evolution of integration in manufacturing automation, *IEEE Industrial Electronics Magazine*, 2007, vol. 1, no. 1, pp. 10–19.
4. **Faldella E., Tucci P.** A model-driven approach to automated diagnosis of industrial distributed I/O systems based on fieldbus technologies *Information, Communication and Automation Technologies: XXII International Symposium on*, Bosnia, 2009, ICAT (2009), pp. 1–7.
5. **Lachello L.** et al. System Comparison, *The 5 Major Technologies Industrial Ethernet Facts*, Iss. 2, Feb., 2013.
6. **Industrial Automation Open Networking Alliance and Luder A.** IAONA Handbook — Industrial Ethernet: (with a List of Producers), IAONA, 2005.
7. **Denisenko V.** Protocoli i seti Modbus i Modbus TCP (Protocols and networks Modbus and Modbus TCP), *STA*, 2000, no. 4, pp. 90–94.
8. **Stouffer K., Falco J., Scarfone K.** Guide to industrial control systems (ICS) security, *NIST Special Publication*, 2011, vol. 800, no. 82, pp. 16.
9. **Mishra A., Mishra D.** Software project management tools: a brief comparative view, *ACM SIGSOFT Software Engineering Notes*, 2013, vol. 38, no. 3, pp. 1–4.
10. **Hannelius T., Salmenpera M., Kuikka S.** Roadmap to adopting OPC UA, *6th IEEE International Conference on Industrial Informatics*, Daejeon, 2008, pp. 756–761.
11. **Boyer S. A.** Scada: Supervisory Control and Data Acquisition, *International Society of Automation*, USA, 2009.
12. **Brodie L.** Thinking Forth, Punchy Pub, 2004.
13. **Riedel S., Bošnjak M., Rocktäschel T.** Programming with a Differentiable Forth Interpreter, arXiv preprint arXiv:1605.06640, 2016.
14. **De Ceballos F.** A Minimal Development Environment for the AVR Processor, *17th EuroForth Conference*, Schloss Dagstuhl, 2001.
15. **Bowman J., Garage W.** J1: a small Forth CPU Core for FP-GAs, *EuroForth 2010 Conference Proceedings*, 2010, pp. 43–46.
16. **Basok B. M., Zakharov V. N., Frenkel S. L.** About estimation of software tests sensitivity to transient faults, *Russian Technological Journal*, 2016, vol. 4, no. 2 (11), pp. 39–44.
17. **Kholopov V. A., Golubov I. N.** Klassifikacija avtomatizirovannyh proizvodstv dlja opredelenija urovnja i metoda ih avtomatizacii (Classification of automated production to determine the level and method of automation), *Polzunovskij Vestnik. Electronic Version*, 2012, vol. 1, pp. 315–317.

Издательство «НОВЫЕ ТЕХНОЛОГИИ»

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала: (499) 269-5397, тел./факс: (499) 269-5510

Технический редактор *Е. В. Конова*. Корректор *Е. В. Комиссарова*.

Сдано в набор 01.11.2016. Подписано в печать 07.12.2016. Формат 60×88 1/8. Бумага офсетная.
Усл. печ. л. 8,86. Заказ МН117 Цена договорная.

Журнал зарегистрирован в Комитете Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций
Свидетельство о регистрации ПИ № 77-11648 от 21.01.02

Учредитель: Издательство "Новые технологии"

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.