

## ВЕРСИОНИРОВАНИЕ И КОНТРОЛЬ ЦЕЛОСТНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРИ УПРАВЛЕНИИ МНОГОКООРДИНАТНЫМИ СИСТЕМАМИ РЕАЛЬНОГО ВРЕМЕНИ

маг. <sup>1</sup>Марко А.Ф., маг. <sup>1</sup>Манин А.С., маг. <sup>1</sup>Нестеренко В.Н.

<sup>1</sup>УО «Белорусский государственный университет информатики и радиоэлектроники», Минск

**Введение.** Объединение узлов точной механики с электронными, электрическими и компьютерными компонентами позволило осуществлять проектирование и производство качественно новых модулей, систем и машин с их интеллектуальным управлением [1]. С развитием электрических приводов и возможностей их применения в индустриально-производственных и транспортных системах, стала очевидна необходимость полной интеграции составляющих элементов электропривода: механики, электрических машин, силовой электроники, микропроцессорной техники и программного обеспечения для наиболее полного использования возможностей современного электропривода, и построения на его основе мехатронных систем перемещения [2]. Можно выделить следующие основные компоненты мехатронной системы прецизионных перемещений: объект перемещения, электропривод (с обратными связями либо без них), драйвер электропривода, интеллектуальный контроллер, аппаратно-программное обеспечение системы управления, пользовательский интерфейс.

Традиционные системы перемещений используют ПЛК-технологии для выполнения задачи управления и включают в свой состав следующие аппаратные и программные элементы: ПК для визуализации, ПЛК с различными сопроцессорами, ввод/вывод через полевую шину, управление движением через параллельный интерфейс LPT, операционную систему и различные языки программирования.

Проведённый анализ современных программно-аппаратных средств показал, что наиболее эффективной технологией для реализации управления системами многокоординатных перемещений в режиме реального времени является технология EtherCAT, внедрение которой требует разработки дополнительных программных приложений [1].

Разработка программного обеспечения требует применения специальных инструментов, которые повышают эффективность разработки за счёт снижения трудоёмкости выполняемых операций. К таким инструментам относятся различные среды разработки общего программного обеспечения и программных средств систем управления [3].

В рамках настоящей работы рассматривается программное обеспечение для версионирования и контроля целостности при управлении системами многокоординатных перемещений в режиме реального времени. Разработка данного программного обеспечения выполнялась в интегрированной среде разработки Visual Studio и комплексной системе управления версиями Team Foundation Server [4].

**Мехатронная система параллельной кинематики на шестикоординатном гибридном приводе.** Для управления в реальном времени всё большее распространение получает технология EtherCAT. Типовая схема пересылки данных в EtherCAT-сети представлена на рисунке 1.

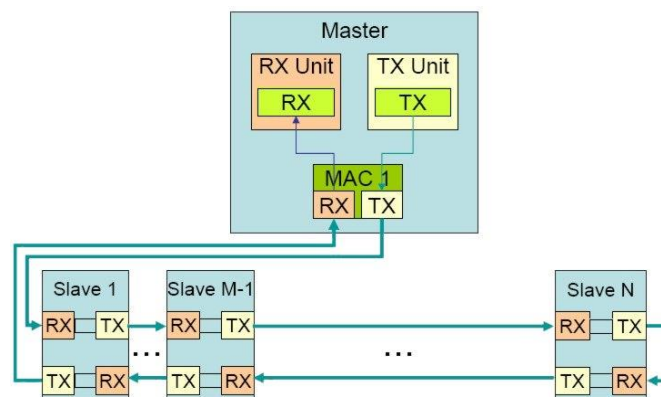


Рис. 1. Типовая схема пересылки данных в EtherCAT-сети

Из всех устройств, подключенных к шине EtherCAT, только мастер может быть инициатором телеграмм. Все остальные устройства «на лету» модифицируют проходящую через них телеграмму, читая и записывая в нее данные технологического процесса. Аппаратная задержка на прохождение телеграммы через одно slave-устройство составляет всего несколько наносекунд.

Таким образом, EtherCAT-технология предоставляет разработчикам систем управления технологическими процессами и сложным оборудованием полностью интегрированное решение, обеспечивающее стандартную и надежную сеть обмена управляющей информацией. При этом количество задействованных полевых шин и интерфейсов уменьшается, обеспечивая тем самым унификацию всех процессов управления, гибкость структуры при практически неограниченном количестве устройств и малое время реакции на события, а также обеспечивается возможность переконфигурирования системы управления без необходимости ее полного отключения.

В связи с наметившимся внедрением технологии EtherCAT в прецизионное технологическое оборудование актуальной и важной является разработка специальных инструментов, позволяющих разрабатывать программное обеспечение системы управления в множестве версий и тем самым с постоянным изменением кода. Поэтому актуальной и важной является автоматизация процесса версионирования такого программного обеспечения и контроль целостности в процессе его эксплуатации.

**Программное средство версионирования и контроля целостности.** При разработке и эксплуатации программного обеспечения (ПО) для систем перемещений важной задачей является обеспечение их цельности, необходимой для предотвращения незапланированных изменений. Контроль за целостностью в предложенном ПО обеспечивается на этапе разработки с помощью внедрения процесса версионирования в интегрированную среду разработки Visual Studio (VS) и систему управления версиями Team Foundation Server (TFS), а на этапе эксплуатации – через формирования и сравнение контрольных сумм. В работе решена задача по разработке программного средства, позволяющего внедрить процесс версионирования в среду VS и систему TFS, контролировать целостность ПО в процессе его эксплуатации.

Для этого было предложено выделить два функциональных модуля. Основные функциональные блоки модулей и их взаимосвязи приведены на рисунке 2.

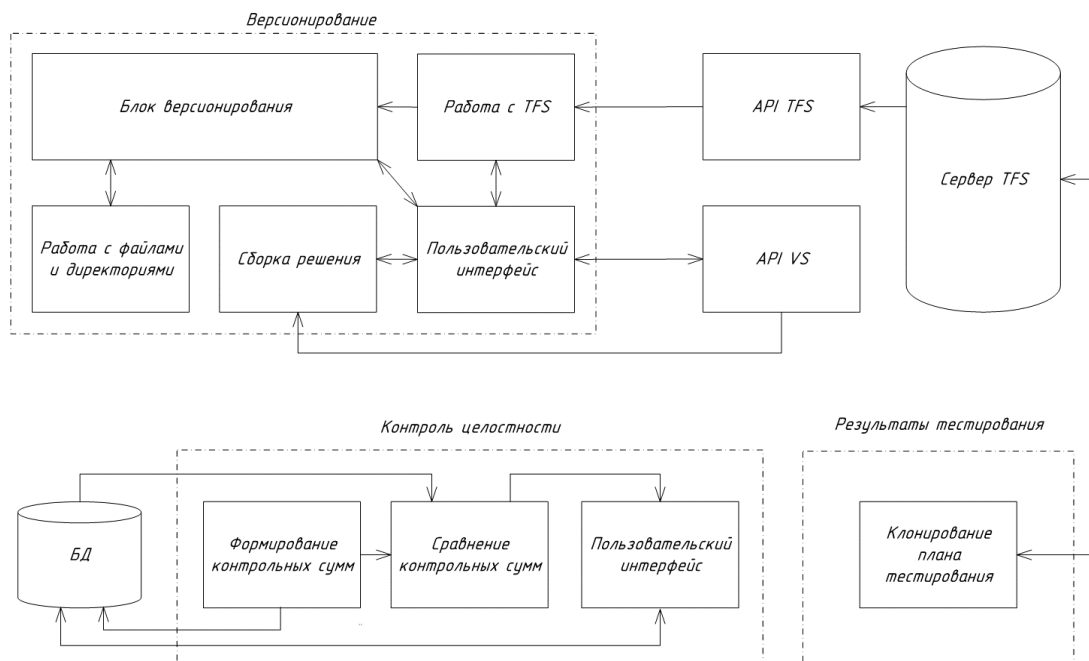


Рис. 2. Функциональные блоки модулей и их взаимосвязи

Модуль версионирования реализован в виде расширения VS и позволяет обновлять версии ПО при внесении изменений. В процессе разработки была реализована концепция формирования версий для проектов, как принадлежащих к версионизируемому решению, так и для проектов, подключённых из других решений по ссылке. В свою очередь проекты, принадлежащие к версионизируемому решению, подразделяются на основные проекты, которые являются источником версии последнего релиза и обычные проекты. Определение типа проекта выполняется с помощью структурного анализа файла решения и конфигурационных файлов. Версия проекта состоит из двух частей: ручной части (первые три старших разряда версии), определяемой последней версией релиза и автоматической части, соответствующей номеру сохранения в системе TFS, в котором был изменён проект. Каждый проект содержит текстовый файл AssemblyInfo, который хранит версию проекта. Для её получения или изменения используются регулярные выражения. В результате после сборки релиза будут сформированы исполняемые файлы, в которых сохранится номер актуальной версии.

Модуль контроля целостности предназначен для определения незапланированных изменений данных в процессе эксплуатации. Для обеспечения целостности данных в процессе эксплуатации необходимо учитывать, что некоторые данные не могут оставаться неизменными, поэтому выделяются части, которые должны быть неизменными в процессе эксплуатации, и части, которые могут изменяться.

Программное обеспечение системы управления состоит из множества различных объектов, таких как исполняемые файлы, файлы данных и объекты баз данных. Формирование контрольных сумм выполняется для каждого типа по-разному. На рисунке 3 приведены выделяемые типы и многоступенчатость процесса формирования контрольных сумм для каждого типа.

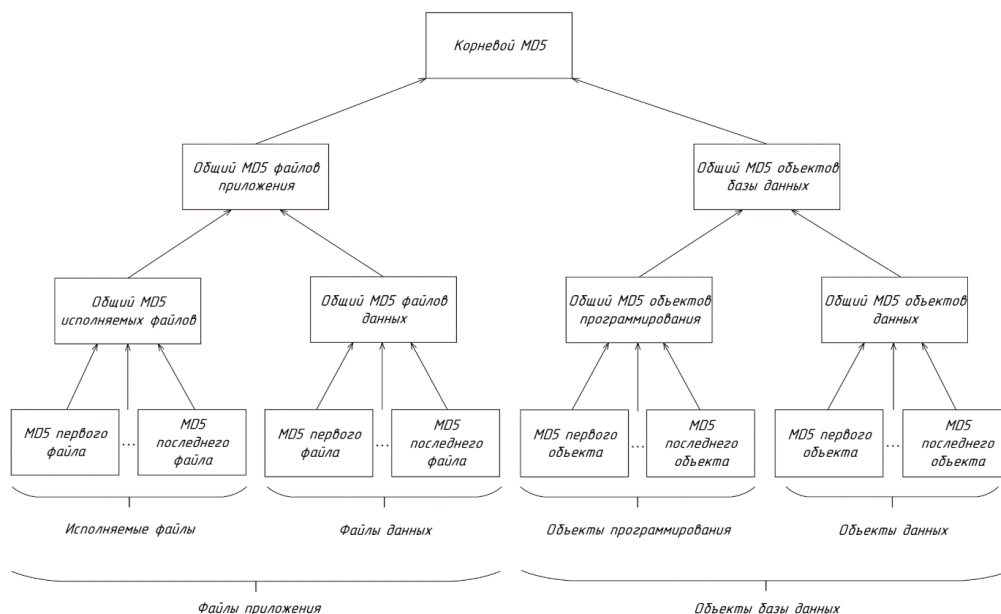


Рис. 3. Многоступенчатый процесс формирования контрольных сумм

Основная цель многоступенчатого формирования контрольных сумм заключается в удобстве представления информации о состоянии каждой подгруппы и в повышении скорости нахождения изменённых файлов.

Таким образом, результатом работы является программное средство, которое контролирует целостность ПО системы управления в процессе разработки и эксплуатации.

**Реализация в тестере печатных плат.** Традиционная аппаратно-программная реализация не позволяет создавать системы управления тестерами печатных плат в реальном режиме времени, поэтому нами была предложена новая аппаратно-программное решение, основанное на технологии EtherCAT. Разработанное программное средство было использовано для обеспечения целостности ПО многокоординатной планарной системы, разработанной и произведенной предприятием ООО «Рухсервомотор» для канадской фирмы Acculogic, выпускающей автоматические установки тестирования печатных плат, предназначенные для реализации финишного контроля печатных плат в сборе с электронными компонентами. Структура EtherCAT-сети для управления шестью планарными позиционерами тестера печатных плат показана на рисунке 4.

Разработанная EtherCAT-сеть содержит один управляющий компьютер Master и шесть локальных систем управления, каждая из которых работает в режиме «управляемый» и обеспечивает реализацию прецизионных перемещений соответствующего планарного позиционера по двум координатам.

Конструктивно многокоординатная система состоит из двух параллельно установленных планарных статоров, расположенных рабочими поверхностями друг к другу. На каждом из статоров, в свою очередь, располагается три планарных якорей с тестирующими иглами и камерами распознавания. Суммарное число степеней свободы такой координатной системы равно 12, при этом по каждой из координат обеспечивается точность перемещений  $\pm 10$  мкм, повторяемость 3 мкм, скорость перемещения до 0,5 м/с и ускорение до 20 м/с<sup>2</sup>. Печатная плата фиксируется в зоне загрузки в плоскости между двумя статорами установки, при этом манипуляторы находятся с обеих сторон печатной платы, тем самым обеспечивая контактирование зондов с верхней и нижней сторон печатной платы. Каждый из зондов обладает шестью степенями свободы, позволяя осуществить подвод контактирующего зонда к контактной площадке под различными углами.

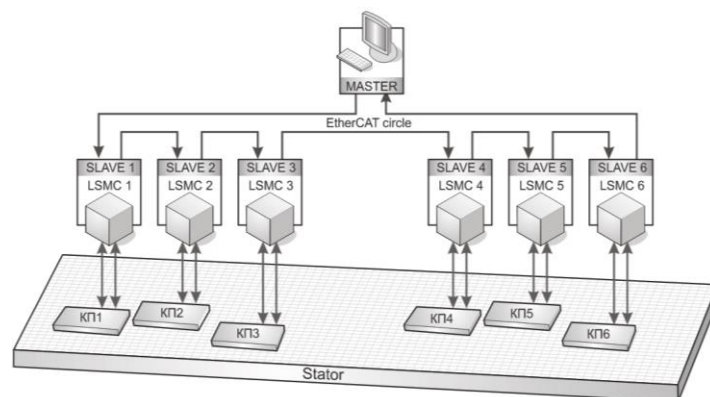


Рис. 4. EtherCAT-сеть для управления шестью позиционерами

Модульное построение системы перемещений обеспечивает при необходимости аппаратно-программное мультиплицирование системы управления, и тем самым быстрое изменение количества используемых зондов.

#### ЛИТЕРАТУРА

1. Карпович, С.Е. Системы многокоординатных перемещений на механизмах параллельной кинематики : монография / С.Е. Карпович [и др.]; под ред. проф. С.Е. Карповича. – Минск : Бестпринт, 2017. – 254 с.
2. Аваков, С.М. Система перемещений для оптико-механического оборудования микроэлектроники / С.М. Аваков, В.В. Жарский, С.А. Русецкий // Доклады БГУИР. – 2007. – № 6. – С. 39–43.
3. Raymont, E. S. The Art of Unix Programming / E. S. Raymont. – Addison-Wesley, 2003 – 547.
4. Chowdhury, K Mastering Visual Studio 2017 // K. Chowdhury / Birmingham B3 2PB, UK, July 2017 – 433 P.