

Боровиков С.С. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ТРИАНГУЛЯЦИОННОЙ ТРЕХМЕРНОЙ СКАНИРУЮЩЕЙ СИСТЕМЫ НА ЯЗЫКЕ C++	277
Волковская А.Г. РАЗРАБОТКА КОНСТРУКЦИИ ИЗМЕРИТЕЛЬНОГО ВОЛНОВОДНОГО ТРАКТА	279
Воропаева В.В. ПРИМЕНЕНИЕ ФОТОЧУВСТВИТЕЛЬНЫХ ИНСТРУМЕНТОВ ПРИ КОНСТРУИРОВАНИИ ПРОСТРАНСТВЕННЫХ СИСТЕМ	282
Закорюкина М.А. ПРИМЕНЕНИЕ МЕТОДА РЕГУЛЯРИЗАЦИИ ПРИ КООРДИНАТНЫХ ИЗМЕРЕНИЯХ ПАРАМЕТРОВ ДУГ ОКРУЖНОСТЕЙ	285
Захарова Т.С. ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНАЯ СИСТЕМА ДЛЯ КОНТРОЛЯ ЛИНЕЙНЫХ ПЕРЕМЕЩЕНИЙ НА БАЗЕ ДАТЧИКОВ МАГНИТНОГО ПОЛЯ	288
Карайкоз М.В. РАЗРАБОТКА АВТОМАТИЧЕСКОЙ ИИС КОНТРОЛЯ ТЕМПЕРАТУРЫ ИЗДЕЛИЯ 1823	290
Королев И.Ю. РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА (ПАК) ДЛЯ ПОДБОРА ТИПОРАЗМЕРА ПРОТЕЗА ТАЗОБЕДРЕННОГО СУСТАВА ПО РЕЗУЛЬТАТАМ ИЗМЕРЕНИЯ И СОВМЕЩЕНИЯ 3D ОБРАЗОВ КАНАЛА БЕДРЕННОЙ КОСТИ И ПРИСОЕДИНИТЕЛЬНОЙ ЧАСТИ ПРОТЕЗА	293
Левкина К.С. ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ ПРИМЕНЕНИЯ НАВИГАЦИОННЫХ ИЗМЕРИТЕЛЬНЫХ СИСТЕМ НА БАЗЕ ТЕХНОЛОГИИ GPS	296
Маляшова А.П. ШЕРОХОВАТОСТЬ ПОВЕРХНОСТЕЙ ХРУПКИХ НЕМЕТАЛЛИЧЕСКИХ МАТЕРИАЛОВ	298
Решетникова Е.А. ИССЛЕДОВАНИЕ ВОПРОСОВ ИЗМЕРЕНИЯ ТРЕХМЕРНОЙ ШЕРОХОВАТОСТИ ЩУПОВЫМ МЕТОДОМ	302
Тверитнев И.О. КОНТРОЛЬ ОТКЛОНЕНИЙ ФОРМЫ ВНУТРЕННИХ СФЕРИЧЕСКИХ ПОВЕРХНОСТЕЙ ПРИ СБОРКЕ ...	305
Терехов Д.В. КОНТРОЛЬ КАЧЕСТВА ПОВЕРХНОСТИ И ГЕОМЕТРИЧЕСКИХ ПАРАМЕТРОВ ТОНКОЙ ПЛАСТИНКИ ИЗ ОПТИЧЕСКОГО СТЕКЛА	308
Шалов О.А. БЕСКОНТАКТНАЯ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНАЯ СИСТЕМА ДЛЯ ИЗМЕРЕНИЯ БОЛЬШИХ РАЗМЕРОВ	310
Ястребова Т.С. ОЦЕНКА НЕОПРЕДЕЛЕННОСТИ ИЗМЕРЕНИЙ ДЛЯ ЛЮКСМЕТРА TESTO 540	313

СЕКЦИЯ «АВТОМАТИЗАЦИЯ И УПРАВЛЕНИЕ»

РЕАЛИЗАЦИЯ УПРАВЛЯЮЩИХ ПРОГРАММ ДЛЯ ТЕСТИРОВАНИЯ НАДЕЖНОСТИ СИСТЕМ ЧПУ (НА ПРИМЕРЕ СИСТЕМЫ ЧПУ АКСИОМА КОНТРОЛ)

Алишынбаева Д.А.

Научный руководитель: Нежметдинов Р.А. — к.т.н., доцент

Кафедра «Компьютерные системы управления» ФГБОУ ВПО МГТУ «Станкин»

Регрессионное тестирование — собирательное название для всех видов тестирования программного обеспечения, направленных на обнаружение ошибок в уже протестированных участках исходного кода. Такие ошибки — когда после внесения изменений в программу перестает работать то, что должно было продолжаться работать, — называют *регрессионными ошибками* (англ. *regressionbugs*).

Регрессионное тестирование выполняется для того чтобы определить как новый функционал повлияет на работу существующей системы. Не приведет ли он к появлению ошибок, сбоев, остановке критичных для бизнеса процессов.

Любая доработка программного обеспечения может повлиять на стабильную работу уже внедренной системы. Предсказать уровень этого влияния и его последствия крайне сложно.

Регрессионное тестирование включает *new bug-fix* — проверка исправления вновь найденного дефекта, *old bug-fix* — проверка, что исправленный ранее и верифицированный дефект не воспроизводится в системе снова, а также *side-effect* — проверка того, что не нарушилась работоспособность работающей ранее функциональности, если её код мог быть затронут при исправлении некоторых дефектов в другой функциональности. Обычно используемые методы регрессионного тестирования включают повторные прогоны предыдущих тестов, а также проверки, не попали ли регрессионные ошибки в очередную версию в результате слияния кода.

Из опыта разработки ПО известно, что повторное появление одних и тех же ошибок — случай достаточно частый. Иногда это происходит из-за слабой техники управления версиями или по причине человеческой ошибки при работе с системой управления версиями. Но настолько же часто решение проблемы бывает «недолго живущим»: после следующего изменения в программе решение перестаёт работать. И наконец, при переписывании какой-либо части кода часто всплывают те же ошибки, что были в предыдущей реализации.

Поэтому считается хорошей практикой при исправлении ошибки создать тест на неё и регулярно прогонять его при последующих изменениях программы. Хотя регрессионное тестирование может быть

выполнено и вручную, но чаще всего это делается с помощью специализированных программ, позволяющих выполнять все регрессионные тесты автоматически. В некоторых проектах даже используются инструменты для автоматического прогона регрессионных тестов через заданный интервал времени. Обычно это выполняется после каждой удачной компиляции (в небольших проектах) либо каждую ночь или каждую неделю.

Регрессионное тестирование является неотъемлемой частью экстремального программирования. В этой методологии проектная документация заменяется на расширяемое, повторяемое и автоматизированное тестирование всего программного пакета на каждой стадии процесса разработки программного обеспечения.

Регрессионное тестирование может быть использовано не только для проверки корректности программы, часто оно также используется для оценки качества полученного результата. Так, при разработке компилятора при прогоне регрессионных тестов рассматривается размер получаемого кода, скорость его выполнения и время компиляции каждого из тестовых примеров.

Пример по тестированию на Аксиоме Контроля приведен ниже:

```

/*
Версия 1.0.
Аксиома Контроль*/
//погрешность измерения
doubleprecision = 0.1;
//pointposition
double vX0, vX1, vX, vY0, vY1, vY, vF;
N10 MSG ("autotest G91")
//-----
// Задаем абсолютную систему координат
N20 G90
// Переходим в начальную точку (X60 Y80)
vX0 = 60;
vY0 = 80;
N30 G00 X=vX0 Y=vY0
// Проверяем достигнута ли точка A по оси X
if (fabs(@AXPOS_X - vX0) > precision)
{
  terminate(2000, "Error. Point A is not valid by axis X");
  goto lb_exit;
}
else
{
  MSG ("Point A valid by axis X")
}
lb_exit:
// Конец

```

N80 M30

Конечный вид экрана оператора с визуализацией отработанной программы показан на рисунке 1.

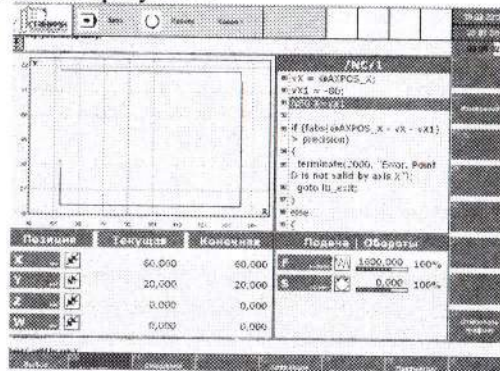


Рис. 1. Окно экрана оператора в автоматическом режиме (в виде графики)

Заключение

Регрессионное тестирование — дорогостоящий род деятельности: процесс регрессионного тестирования может включать исполнение достаточно большого количества тестов на скорректированной программе, даже если изменений очень мало. Несмотря на то, что усилия, требуемые для внесения небольших изменений, как правило, минимальны, они могут требовать достаточно больших усилий для проверки качества изменённой программы. Тем не менее, проведение регрессионного тестирования необходимо. Надёжная и эффективная разработка и сопровождение программного обеспечения невозможна без регрессионного тестирования.

Библиографический список:

1. Ловыгин А.А., Васильев А.В. - "Современный станок с ЧПУ и CAD/CAM система" // Учеб.пособие. – М., 2013г.
2. Рыбников С.В. «Руководство программиста по созданию управляющих программ AxiOMA Ctrl v. 0.1» СТ00.041600.001 ЭП
3. Сосонкин В.Л., Мартинов Г.М. Системы числового программного управления: Учеб.пособие. – М. Логос, 2005. – 296 с. ISBN 5-98704-012-4.
4. Шемелин В.К., Нежметдинов Р.А. Повышение качества архитектурных решений систем ЧПУ на основе программно реализованного контроллера типа SoftPLC // Автоматизация и современные технологии, №6, 2008. с. 33-35, Машиностроение, Москва
5. Нежметдинов Р.А., Кулиев А.У., Николушкин А.Ю., Червоннова Н.Ю. Управление электроавтоматикой токарный и токарно-

фрезерных станков на базе Soft PLC // Автоматизация в промышленности, №4, 2014. с. 49-51.

6. Елифанов Н.А. Методы реализации регрессионного тестирования по расширенным тестовым набором // Санкт-Петербург, 2003.

ПРИМЕНЕНИЕ ОТЕЧЕСТВЕННЫХ МИКРОКОНТРОЛЛЕРОВ ДЛЯ УПРАВЛЕНИЯ МЕХАТРОННЫМИ УСТРОЙСТВАМИ

Арсентьев А.И.

*Научный руководитель: Илюхин Ю.В. – д.т.н., профессор
Кафедра «Робототехника и мехатроника» МГТУ «СТАНКИН»*

В настоящее время следящие системы являются важной основой функционирования большого числа технических устройств, применяемых в различных сферах робототехники и машиностроения. К следящим системам можно отнести: приводы систем автоматического регулирования, устройств пуска, рулевые и приборные приводы и многое другое. Повышение качества регулирования и улучшение технико-экономических показателей подобных устройств требует постоянного совершенствования систем управления процессами, происходящими в них. Кроме того, для российских предприятий часто является актуальным и необходимым использование отечественных электронных компонентов. В настоящее время в России прослеживается проблема, связанная с небольшим размером отечественной электронной элементной базы для применения в различных цифровых системах управления, радиосвязи и изделиях, требующих точные аналогово-цифровые и цифро-аналоговые преобразования, а значит и проблема выбора элементов отечественного производства для использования в области цифрового управления.

Часто, выбор микроконтроллера является одной из самых приоритетных задач, от которой зависит успех проекта в целом. Важно выбрать наиболее экономически выгодный вариант, удовлетворяющий поставленным требованиям по производительности, надежности, условиям применения и т.д. Можно выделить основные критерии выбора микроконтроллера:

1. Пригодность МК, т.е. возможность реализации системы на основе однокристалльного МК.
2. Наличие необходимых контактов, портов ввода/вывода.
3. Наличие необходимых периферийных устройств.
4. Производительность ядра.
5. Доступность программного обеспечения и наличие средств отладки.

Одним из перспективных вариантов отечественных микроконтроллеров для построения современных систем компьютерного управления является 16-разрядный микроконтроллер 1874BE96T

отечественного производства, разработанный фирмой ООО «НИИЕТ». МК представляет собой сверхбольшую интегральную схему однокристалльного 16-разрядного МК с тактовой частотой до 33 МГц, имеет ОЗУ (2024×8) бит, расширенное ОЗУ (2048×8) бит, восемь 16-разрядных АЦП с возможностью дифференциального включения входов и блоком цифровых компараторов, 14-разрядный ЦАП, 3-канальный ШИМ, два последовательных порта ввода-вывода UART, синхронный последовательный интерфейс SPI, интерфейс I2C, с внутренней памятью программ типа EEPROM (16K×16) бит, устройство высокоскоростного ввода-вывода, отладочный модуль и сторожевой таймер.

В результате применения микросхемы возможно достичь увеличения функциональности, надежности и производительности встроенных систем управления (в сравнении с микроконтроллерами на базе MCS-96) за счёт использования полностью нового быстродействующего 16-разрядного ядра, высокоточных 16-разрядных АЦП и 14-разрядного ЦАП, большого объема внутрикристалльного ОЗУ (4 Кбайт), энергонезависимой памяти программ типа EEPROM (32 Кбайт), устройства высокоскоростного ввода-вывода, четырех последовательных портов приема/передачи. Также, благодаря использованию встроенных ЦАП и АЦП возможно упростить устройство, уменьшить размер и стоимость периферийных компонентов, подключаемых к МК.

В МК реализована, архитектура «Регистр-Регистр». Такая архитектура обеспечивает достижение наиболее высокой производительности и упрощает работу с периферией. Этим, в первую очередь, объясняется возможность эффективного решения сложных задач управления в реальном времени. В основе ядра микроконтроллера лежит регистровый файл. Большое число универсальных легкодоступных регистров исключает «узкие места», свойственные архитектуре, использующей специальные регистры-аккумуляторы, и обеспечивает быстрое переключение контекста. Все устройства МК реализуют операции с байтами, словами, несколько операций с 32-битовыми операндами, а также команды перехода по битам.

Основной функцией МК является обеспечение управления устройствами в реальном времени. Схема контроллера прерываний позволяет событиям реального времени управлять выполнением программы. Когда событие вызывает прерывание, ядро обслуживает это прерывание перед выполнением следующей команды. Встроенная периферия, внешний сигнал или команда могут выставить запрос на прерывание. В разработанном микроконтроллере – 44 источника прерывания и 17 векторов прерывания.

Архитектура МК направлена на создание цифровых управляющих систем, способных функционировать в системе реального времени, может быть адаптирована и модифицирована под конкретные приложения. Наличие встроенных аппаратных сдвигателя, умножителя и делителя и