

УДК 621.01

СИСТЕМА УПРАВЛЕНИЯ ОРТОГОНАЛЬНЫМ МОДУЛЬНЫМ РОБОТОМ УМР-3-0 И ЕЕ ПРИМЕНЕНИЕ В УЧЕБНОМ ПРОЦЕССЕ

Карпов Алексей Александрович⁽¹⁾

*Студент 3 курса⁽¹⁾, кафедры МТ-1 «Металлорежущие станки»,
Московский Государственный Технический Университет им. Н.Э. Баумана*

*Научный руководитель: Вуколов А.Ю.
ассистент кафедры РК-2 «Теория механизмов и машин»,
Московский Государственный Технический Университет им. Н.Э. Баумана*

Введение

Одной из актуальных задач сегодняшней машиностроительной индустрии является возможность выполнения наиболее сложных и высокоточных операций при минимуме затраченных финансовых средств. А также, возможность модернизации старого имеющегося оборудования под выполнение необходимых работ, повышение его точности и функциональности. Достигнуть этого можно путём реализации эффективного взаимодействия мехатроники робота и управляющего персонального компьютера. Кроме того, при преподавании робототехнического раздела курса теории механизмов и машин студентам машиностроительных специальностей существует задача доступно продемонстрировать основные принципы робототехники на конкретных примерах, желательно – реальных устройствах. Кроме того, для повышения эффективности преподавания успевающим студентам – необходимо предоставить возможность самостоятельного создания, отладки и испытания ПО управления робототехническими устройствами. Естественным методом решения подобной задачи является использование имеющегося оборудования. Участие же в ремонте и модернизации реального устройства позволяет студенту приобрести практические навыки работы с робототехническими системами.

В данной работе за основу был взят учебный модульный робот УМР-3-0 с ограниченной функциональностью (неуправляемые по скорости двигатели степеней, с остановом путем закорачивания обмотки возбуждения), для него был разработан комплект ПО, позволивший расширить его возможности до выполнения операций повышенной сложности, и создать гибкую систему, готовую к интеграции в учебный процесс кафедры ТММ. Робот установлен в лаборатории кафедры ТММ и успешно применяется в демонстрационных целях.

Описание конструкции робота УМР-3-0

Робот УМР-3-0 (производство МИРЭА/«Росучприбор») представляет собой конструкцию из трёх мехатронных модулей, собранную с учетом обеспечения движения в прямоугольной системе координат с рабочей зоной, совпадающей по конфигурации с зоной обслуживания. Изначально робот предназначен для исполнения операций конвейерного типа, т.е. перемещения предметов в ограниченном пространстве без препятствий внутри зоны обслуживания. Общий вид робота УМР-3-0, смонтированного на тестировочном столе, представлен на рис.1. Полное описание конструкции мехатронных модулей робота приведено в [1].

На конце модуля, обеспечивающего движение по оси Y, вместо захватывающего элемента был установлен электродвигатель с закреплённой на выходном валу фрезой.

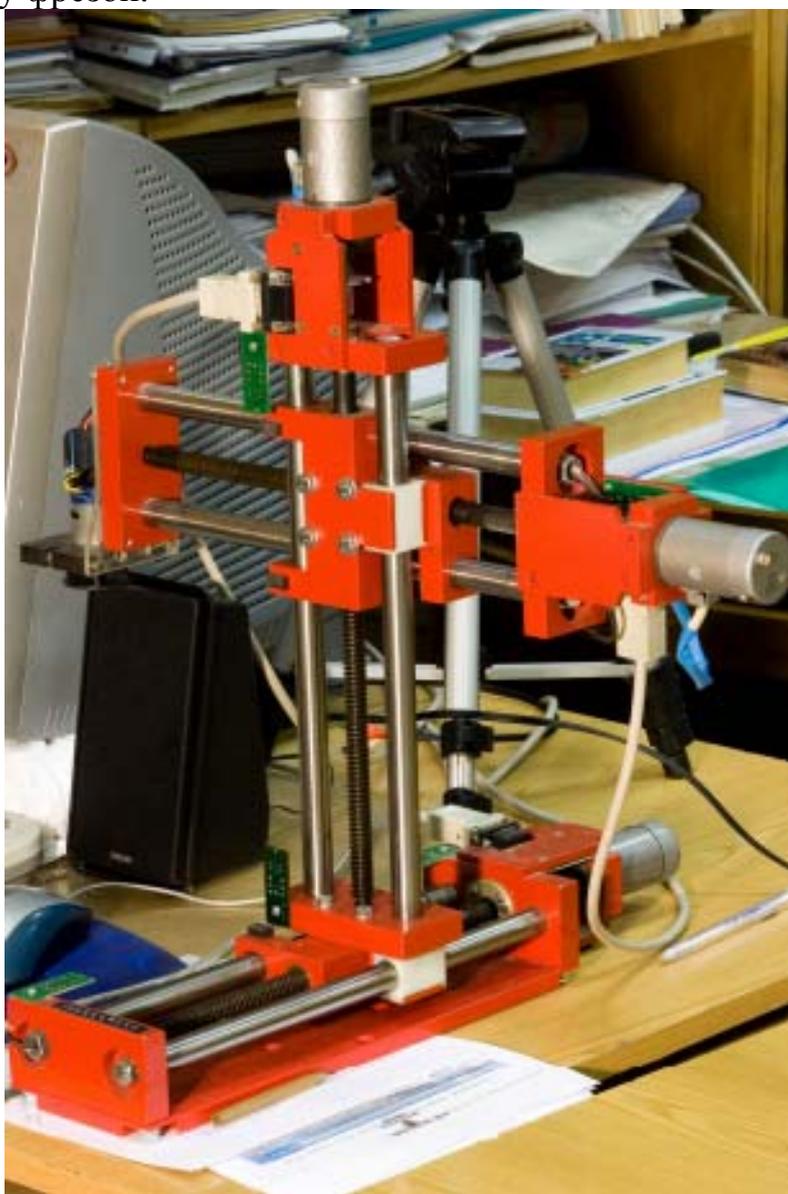


Рис. 1. Робот УМР-3-0

На каждом модуле установлено по три датчика: два из них представляют собой герконовые сигнализаторы границ рабочей зоны, и один – оптронный датчик позиционирования (рис. 2).

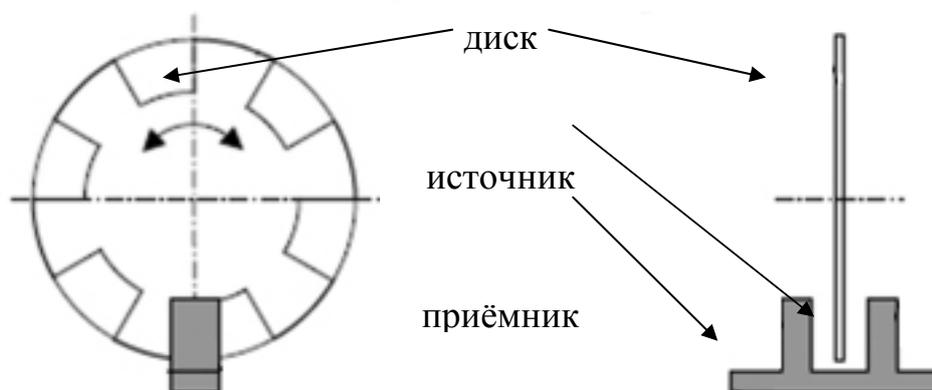


Рис. 2. Датчик позиционирования

Датчик позиционирования представляет собой круглый диск с прорезями, закреплённый на ходовом винте степени. При вращении винта диск перекрывает световой поток в оптронной паре и посылает соответствующие сигналы на блок управления, что позволяет судить о положении соответствующей степени. Частота вращения двигателей робота постоянна и не регулируется при данной конструкции блока управления. Точность позиционирования степеней составляет ± 0.5 мм по координате и ± 10 мм/с по скорости.

Итак, мы имеем малофункциональный механизм, способный перемещать фрезу внутри прямоугольной рабочей зоны, но без возможности установки её в точку пространства с необходимыми точными координатами.

Общий вид консоли блока управления роботом представлен на рис. 3. Особенности блока управления:

- Связь с управляющим компьютером по интерфейсу Centronix/LPT Bidirectional Parallel Interface;
- Эмуляция на LPT-порте системы команд микро-ЭВМ архитектуры 8255;
- Принудительный опрос датчиков без возможности реализации прямой обработки события изменения состояния порта;
- Чтение и запись данных в порте LPT по одним и тем же линиям (с изменением режима работы порта), что требует разработки специальных процедур высокоскоростного опроса датчиков;
- Наличие программируемой линейки светодиодов, служащей для необходимой пользовательской индикации режима работы системы.

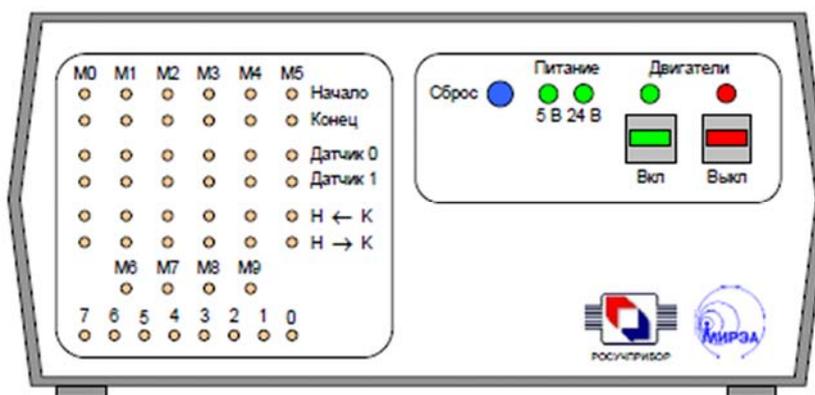


Рис. 3. Общий вид консоли блока управления роботом УМР-3-0

Постановка задачи

Учитывая потребности использования робота УМР-3-0 в учебном процессе, возможно сформулировать задачи разработки системы управления:

1. Создать библиотеку процедур и дополнительные программные средства, позволяющие студенту самостоятельно разрабатывать и тестировать программное обеспечение для управления роботом;

2. Разработать программное обеспечение, реализующее управление роботом на низком уровне. Данное ПО должно обеспечить возможность демонстрации основных принципов мехатроники, управления робототехническими системами и взаимодействия управляющего устройства робота с компьютером. Применение – в лабораторном практикуме по курсу ТММ для студентов всех специальностей [2];

3. Создать специальное ПО, реализующее позиционирование степеней свободы робота с учетом всех особенностей конструкции, синхронизацию реальной машины с ее 3D-моделью в памяти компьютера, что позволит демонстрировать в рамках лабораторного практикума реализацию системы управления четвертого поколения [4, 5].

4. Реализовать пп. 1-3 без модернизации аппаратного обеспечения.

Реализация библиотеки процедур и программы низкоуровневого управления

Основной особенностью стандартной библиотеки процедур, поставляемой в комплекте с блоком управления роботом является прямой доступ к портам. Подобная реализация ПО работает только под управлением операционных систем PC DOS/MS-DOS ниже 6 версии. Однако, реалии учебного процесса требуют наличия Windows-реализации библиотеки процедур с опосредованным вводом-выводом данных в LPT-порт посредством драйвера. Такая реализация была разработана в виде набора открытого исходного кода на языках C++ (разработчик – А. Вуколов) и ANSI Object Pascal/Delphi (разработчик – автор). Основой реализации послужил свободно распространяемый WinNT-драйвер виртуального устройства

прямого доступа к портам WinIO.vxd (автор Я. Каплан) [3]. Тексты библиотеки процедур доступны студентам на кафедре ТММ по запросу, что позволяет предоставить возможность самостоятельной разработки ПО для робота УМР-3-0. Все программы, составленные в рамках данной работы, используют библиотеку процедур в текущей версии.

Программа низкоуровневого управления ROBOEXEC.EXE для робота УМР-3-0 реализует прямой доступ к эмулируемым регистрам микро-ЭВМ архитектуры 8255. Общий вид главного окна программы в английском языковом варианте представлен на рис. 4.

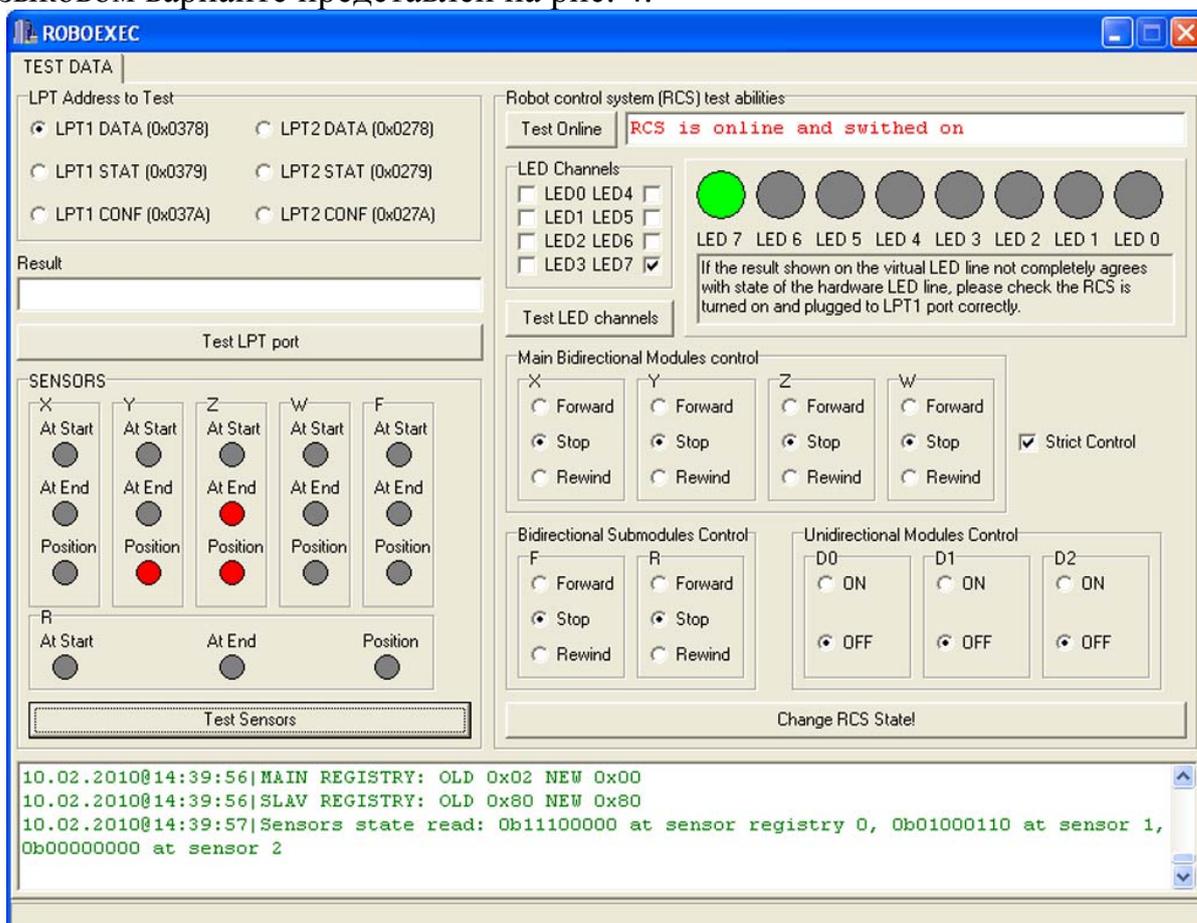


Рис. 4. Общий вид главного окна программы ROBOEXEC.EXE

Программа ROBOEXEC обеспечивает прямой доступ к регистрам блока управления роботом УМР-3-0, подключенного к порту LPT1 или LPT2. Кроме того, предусмотрена возможность подключения к портам блока управления – дополнительных степеней свободы в случае модернизации конструкции робота.

Любое изменение состояния регистров блока управления требует непосредственного воздействия пользователя на виртуальные органы управления программы. Значения, передаваемые в регистры, задаются установкой переключателей, а также протоколируются в специальном окне. Программа не является событийно-ориентированной – опрос датчиков производится однократно по команде пользователя (рис. 5).

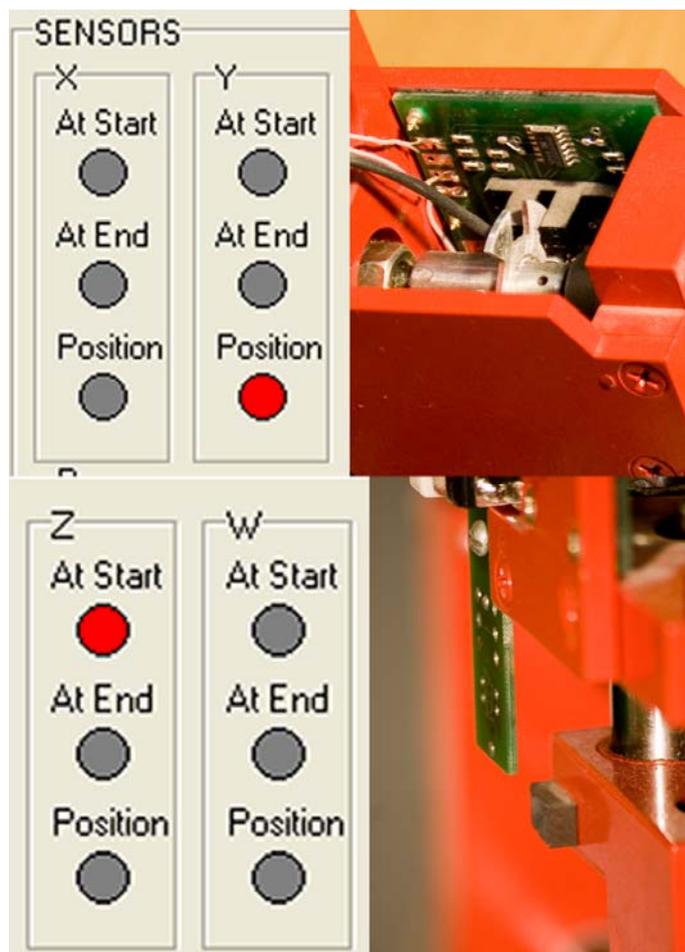


Рис. 5. Работа программы ROBOEXES.EXE с датчиками

В состоянии «ВКЛ» находятся: датчик позиционирования степени свободы по оси “Y” и датчик границы рабочей зоны степени по оси “Z”.

Особенностью ROBOEXES.EXE является возможность управления выбегом двигателей – в случае, если проводится останов одной операцией всех включенных двигателей степеней – переключатель Strict Control обеспечивает возможность такого останова без закорачивания обмотки, что открывает возможности измерения выбега степеней и его влияния на позиционирование. Программа ROBOEXES.EXE успешно применяется в лабораторном практикуме по курсу ТММ.

Особенности создания программы управления

При преподавании робототехнического раздела курса ТММ важной задачей является формирование у студента объективного представления о взаимодействии между реальным робототехническим устройством и виртуальной производственной средой (или средой планирования технологических операций). С учетом данного требования была разработана программа SKAYNET.EXE, в которой реализовано взаимодействие предельно упрощенной 3D-среды планирования и реального

робототехнического устройства. Общий вид главного окна программы представлен на рис. 6.

Программа создана в среде Delphi [6] с применением свободно распространяемой графической надстройки GLScene, упрощающей работу с OpenGL в Delphi [7]. Принцип работы программы основан на одновременном исполнении (в нескольких потоковых объектах Windows) нескольких программных таймеров: часть таймеров отвечает за опрос порта LPT компьютера, к которому подключён робот (съем данных с датчиков), и за отсылку требуемых данных на линии того же порта.

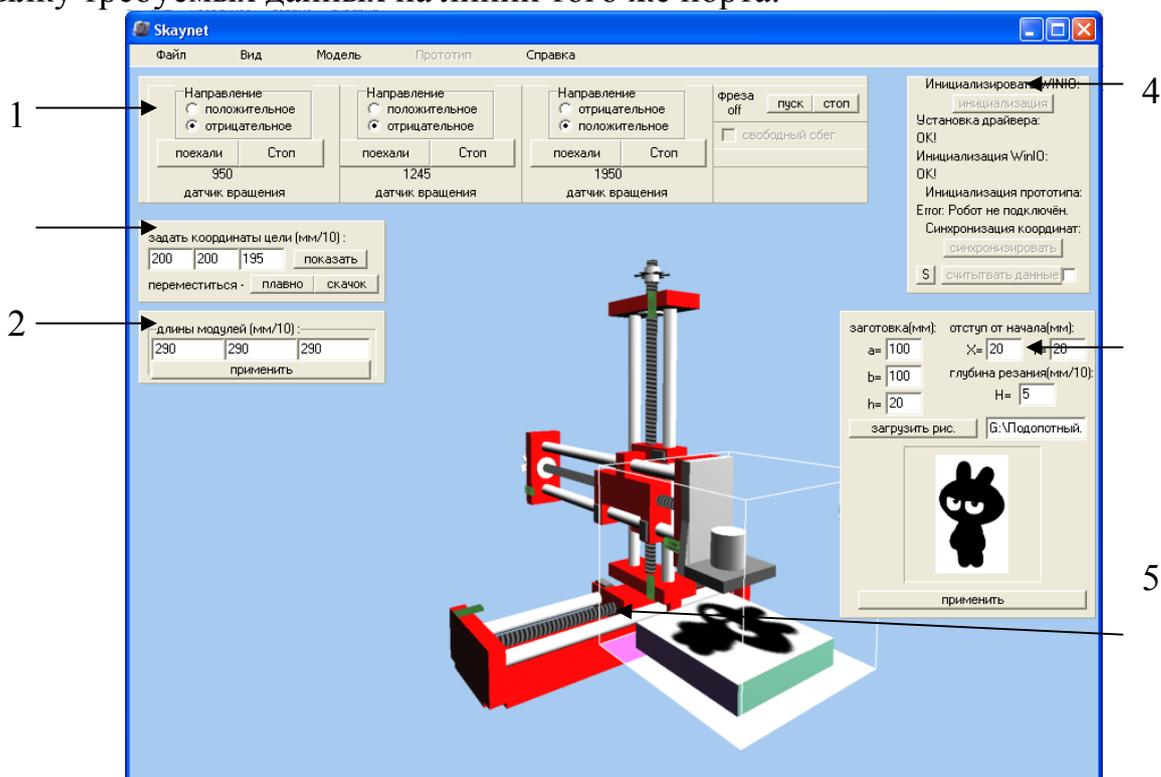


Рис. 6. Общий вид главного окна программы SKAYNET.EXE

1 – главная техническая панель, отвечающая за управление движением модулей робота; 2 – панель установки фрезы; 3 – панель изменения длин модулей для случая нестандартной комплектации робота; 4 – панель синхронизации (автоматической установки системы в начальную позицию); 5 – панель подгрузки 2D-модели изделия; 6 – 3D-модель робота.

Другие таймеры отвечают за задание входных данных аналитического просчёта движения реального робота. Один из таймеров является времязадающим элементом синхронизации анимированной 3D-модели робота, встроенной в программу для наглядности представления происходящего процесса. Кроме того, в программе реализована возможность управления роботом с клавиатуры.

Принцип взаимодействия пользователя с программой и программы с роботом можно представить в виде блок-схемы (рис. 7). Исполнение программы изначально распараллелено по трем потокам: главный поток VCL (объектно-ориентированная реализация управления окнами и потоками Windows в среде Delphi/C++ Builder) отвечает за аналитический просчёт положения модулей робота и выполнения необходимых операций по

изменению их положения; второй поток – за рендеринг 3D изображений; третий поток включает функциональность опроса LPT порта и отсылку на него данных. Движение каждого модуля закреплено за соответствующим программным таймером, запускающимся при начале движения модуля и выключающимся при завершении движения модуля. Для повышения точности позиционирования и увеличения частоты переключения порта вместо стандартного таймера Windows применена реализация таймера WinNT – MCI Timing Engine/DirectX, что позволяет потенциально получить частоту опроса датчиков в текущей реализации библиотеки процедур до 120-150 Гц.

Данные, введённые пользователем в программу, обрабатываются главным потоком VCL, откуда часть соответствующей информации передаётся в поток, отвечающий за прорисовку 3D модели, выводимый на экран; а часть в поток, отвечающий за генерацию и передачу сигнала через LPT порт на блок управления робота в виде байт. Блок управления на основе полученных данных координирует работу двигателей робота. Датчики движения и позиционирования при срабатывании передают сигналы в блок управления, которые, в свою очередь, считываются с LPT-порта по сигналу ещё одного таймера.

Для подтверждения достижения заданной точности позиционирования проводился эксперимент (рис. 8). По полученным результатам можно сделать вывод о том, что погрешность позиционирования рабочего органа робота при текущей реализации системы управления не превышает 1 мм.

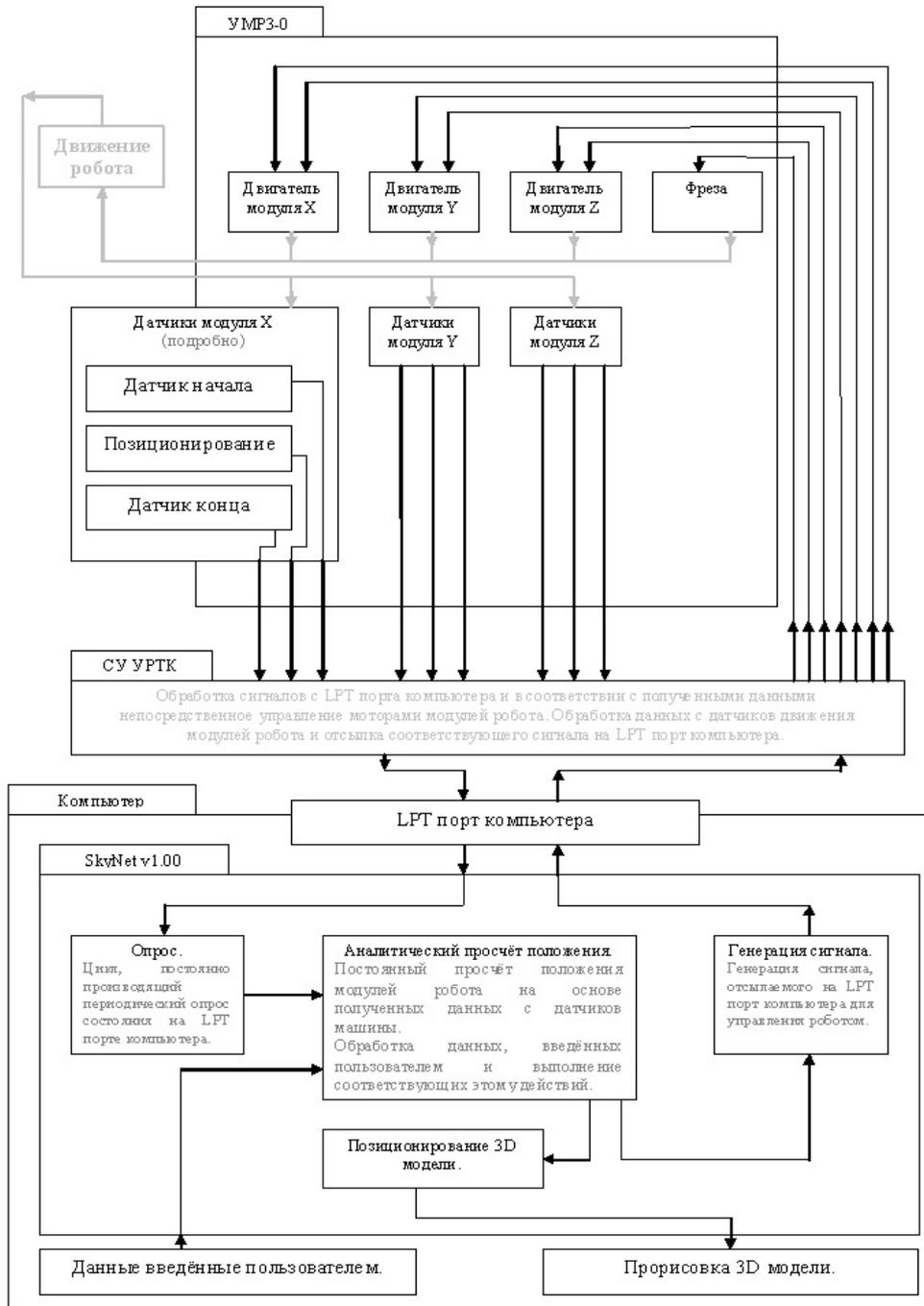


Рис. 7. Блок-схема программно-аппаратного комплекса SKAYNET.EXE + УМР-3-0

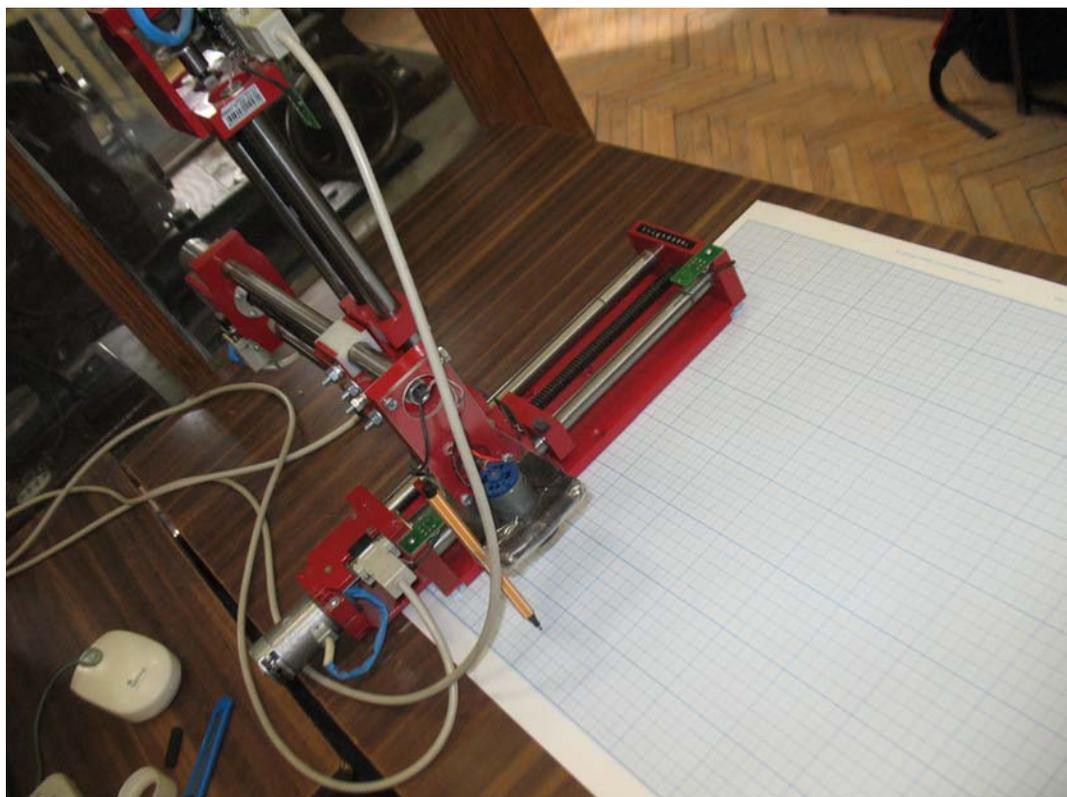


Рис. 8. Эксперимент по проверке точности позиционирования

Заключение

При подготовке данной работы успешно решена задача создания библиотеки процедур управления, обеспечивающей программирование робота УМР-3-0 на языках С++ и ANSI Object Pascal под управлением Windows.

Разработан комплект программного обеспечения, позволяющий наглядно продемонстрировать низкоуровневое (ROBOEXEC.EXE) управление роботом, и реализацию системы управления четвертого поколения с виртуальной средой планирования технологической операции (SKAYNET.EXE).

Внедрение робота УМР-3-0 в связке с программами ROBOEXEC/SKAYNET в учебный процесс (лабораторный практикум) по курсу теории механизмов позволяет существенно повысить эффективность преподавания робототехнического раздела курса. Студенты хорошо воспринимают наглядно демонстрируемые на реальном примере основные принципы робототехники. Дополнительно, успевающим студентам предоставляется возможность самостоятельно реализовывать и проверять авторские алгоритмы управления реальной робототехнической системой. На данный момент учебная робототехническая система установлена в лаборатории кафедры ТММ, оснащена управляющим компьютером и успешно используется в учебном процессе.

Литература

1. Учебные модульные роботы: техническое описание. Росучприбор, 2002 (электронный ресурс)
<http://www.rosuchpribor.ru/russian/Prof2007/robot/umr-2m.html>
2. Тарабарин В.Б., Кузенков В.В., Фурсяк Ф.И. Лабораторный практикум по теории механизмов и машин. М., МГТУ, 2009
3. WinIO.sys help and user guide for Object Pascal and C++ Builder users (2006). Open source. Available from Yariv Kaplan (электронный ресурс)
http://www.internals.com/utilities_main.htm
4. A. Golovin, M. Ceccarelli: Usage of real mechanisms and models in Machines and Mechanisms Theory course trainings: the album of illustrations. Registration Cert. №16335, 21.05.2009/State registration cert. №0320900972
5. A. Golovin, D. Saschenko: Expert Judgment of Mechanical parts of Robotics. Edited by S. Jatsun. Vibration Machines and technology: Proceedings of Scientific and Technical Conference. – Kursk (2008). Pp. 834-841.
6. Фленов М.Е. Библия Delphi. ISBN 978-5-9775-0116-3. С-Пб., «БХВ-Петербург», 2008.
7. Ru.Wikipedia.org: GLScene. Published under GNU Public License (электронный ресурс)
<http://ru.wikipedia.org/wiki/GLScene>