

ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Технология машиностроения»

АВТОМАТИЗАЦИЯ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ

*Методические рекомендации к практическим занятиям
для студентов специальности
1-36 01 01 «Технология машиностроения»
дневной и заочной форм обучения*



Могилев 2018

УДК 65.011.56
ББК 65.050.2
А 22

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Технология машиностроения» «10» октября 2017 г.,
протокол № 2

Составители: канд. техн. наук, доц. В. М. Шеменков;
канд. техн. наук, доц. Е. В. Ильюшина

Рецензент канд. техн. наук, доц. В. В. Кутузов

Даны задания для практических занятий по дисциплине «Автоматизация
производственных процессов», приведены методические указания по их вы-
полнению, перечень необходимой литературы.

Учебно-методическое издание

АВТОМАТИЗАЦИЯ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ

Ответственный за выпуск	В. М. Шеменков
Технический редактор	А. Т. Червинская
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60x84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л . Тираж 56 экз. Заказ №

Издатель и полиграфическое исполнение:
Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 24.01.2014.
Пр. Мира, 43, 212000, Могилев.

© ГУ ВПО «Белорусско-Российский
университет», 2018



Содержание

1 Практическая работа № 1. Разработка структуры управления и компоновки гибких производственных модулей для операции технологического процесса.....	4
1.1 Интерфейс системы автоматизации моделирования и программирования роботизированных технологических комплексов RobotStudio.....	4
1.2 Работа с библиотеками цифровых прототипов технологического оборудования и оснастки.....	6
1.3 Определение движений элементов технологического оборудования и оснастки.....	11
2 Практическая работа № 2. Разработка структуры управления и компоновки ГПС (ГАЛ, ГАУ) для технологических процессов обработки деталей.....	20
2.1 Задание пути перемещения и моделирование движения рабочих инструментов робота.....	20
2.2 Управление входными и выходными сигналами контроллера робота.....	28
2.3 Цифровое прототипирование технологического оборудования РТК и ГПС.....	34
2.4 Общие сведения о контроллере IRC-5 и офлайн-программировании РТК на языке RAPID.....	41
Список литературы.....	46



1 Практическая работа № 1. Разработка структуры управления и компоновки гибких производственных модулей для операции технологического процесса

Разработка структуры управления и компоновки гибких производственных модулей (ГПМ) производится в среде RobotStudio, позволяющей моделировать и программировать роботизированные технологические комплексы (РТК), оснащенные роботами производства концерна **ABB Group** (Швеция).

RobotStudio дает возможность оптимизировать роботизированный технологический процесс на экране компьютера, что позволяет избежать ошибок при программировании РТК и ускоряет их процесс внедрения.

1.1 Интерфейс системы автоматизации моделирования и программирования роботизированных технологических комплексов RobotStudio

После загрузки системы RobotStudio открывается главное окно графического интерфейса пользователя (Graphical User Interface, GUI) системы RobotStudio (рисунок 1.1).

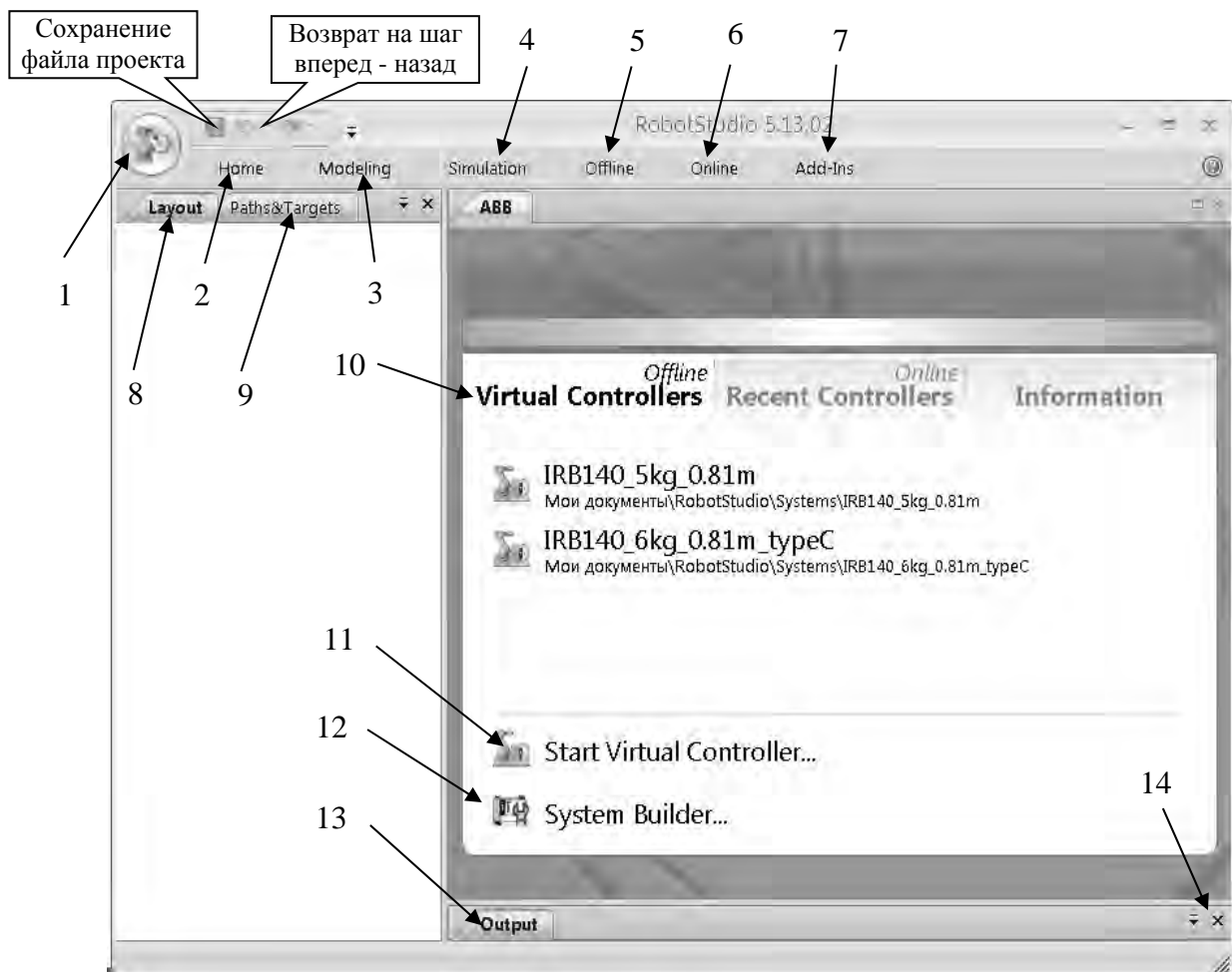



Рисунок 1.1 – Главное окно графического интерфейса пользователя

Главное меню графического интерфейса пользователя состоит из следующих разделов.

1 Кнопка  открывает дополнительное меню для доступа к командам управления файлами, печати и настройки системы (**RobotStudio Options**).

2 **Home** открывает панель инструментов, которая содержит команды для создания роботизированных технологических систем из различных компонентов (**Build Station**), программирования путей инструментов (**Path Programming**), задания рабочих объектов (**Settings**), перемещения объектов и отдельных звеньев робота вручную (**FreeHand**), управления видами в 3D-пространстве (**3D View**).

3 **Modeling** открывает панель инструментов, которая содержит команды для создания 2D- и 3D-компонентов РТК с помощью CAD-операций (**CAD Operations**), выполнения измерений (**Measure**), перемещения объектов и отдельных звеньев робота вручную (**FreeHand**), создание анимированных движений цифровых прототипов механизмов и инструментов (**Mechanism**).

4 **Simulation** открывает панель инструментов, которая содержит команды для настройки, формирования (**Collision**), управления (**Conveyer Tracking**), контроля (**Simulation Control**) процедуры имитации (симуляции) работы роботизированного технологического процесса (**Monitor**) и записи видеоролика (**Record Movie**) имитационной модели технологического процесса для демонстрации заказчику РТК.

5 **Offline** открывает панель инструментов, которая содержит команды для синхронизации (**Synchronize**), конфигурирования виртуального контроллера (VC) (**Virtual Controller**), работы с программными модулями (**RAPID**), написанными на языке программирования **RAPID** для роботов производства ABB Group и конфигурирования проектируемой системы **Configuration**.

6 **Online** открывает панель инструментов, которая содержит команды для подключения (**Access**), настройки (**Controller Tools**) конфигурирования реального контроллера IRC5 (**Configuration**) и работы с программными модулями (**RAPID**), написанными на языке программирования RAPID для роботов производства ABB Group.

7 **Add-Ins** открывает панель инструментов, которая обеспечивает доступ в среду системы программирования **Visual Studio Tools for Application (VSTA)** для расширения функциональных возможностей RobotStudio.

8 **Layout** – закладка в виде иерархической структуры (дерева проекта) для просмотра и изменения свойств компонентов структуры РТК.

9 **Paths & Targets** – закладка в виде иерархической структуры для просмотра и изменения свойств программируемых путей и целей рабочего инструмента робота.

10 **Virtual Controllers** – окно быстрого старта, которое содержит команды:

Recent Stations^{Offline} – загружает недавно созданные рабочие станции, которые приводятся в списке ниже;

Recent Controllers^{Online} - – подключает контроллер, который использовался в последних сеансах работы, устанавливает связь с реальным контроллером;



Information – обеспечивает доступ к справочным файлам, просмотру обучающих программ, управлению лицензиями, просмотру новостей RobotStudio.

11 **Start Virtual Controllers...** – подключение виртуального контроллера.

12 **System Builder...** – создание новой системы.

13 **Output** – окно для вывода информации о состоянии системы и ошибках, которые возникают в процессе моделирования работы системы.

14 **Open-Close Window Output** – открытие и закрытие окна Output.

Индивидуальное задание.

В соответствии с заданным вариантом создать модель РТК, состоящую из робота и платформы для его линейного перемещения.

Вариант 1. Грузоподъемность робота – до 40 кг, длина пути платформы – до 4 м.

Вариант 2. Грузоподъемность робота до – 205 кг, длина пути платформы – до 10 м.

Для выбора платформы следует использовать данные из таблицы 1.1. В ходе выполнения задания необходимо вести протокол построения модели РТК.

Таблица 1.1 – Данные для выбора платформы

IRBT 4004/6004/7004	Robot	Travel length	No of robots	Mounting pos
IRBT 4004	IRB 4400/4450S/4600	1,9...19,9 m / 1 m step	One/track	Floor
IRBT 6004 / 7004	IRB 6620/6640/ 6640ID/6650S	1,7...19,7 m / 1 m step	One/track	Floor
IRBT 7004	IRB 7600	1,7...19,7 m / 1 m step	One/track	Floor

1.2 Работа с библиотеками цифровых прототипов технологического оборудования и оснастки

Создайте станцию без использования шаблона (пункт **New Station** (новая станция)). В левой части окна **New Station** щелчком мыши выберите пункт меню **Empty Station** (создание новой «пустой» станции). В основном окне RobotStudio появится условное изображение площадки размером 2 × 2 м для размещения компонентов РТК (рисунок 1.2).

Для создания модели РТК воспользуйтесь библиотеками цифровых прототипов технологического оборудования и оснастки. Доступ к библиотекам RobotStudio возможен из раздела меню **Home**. В левой части меню кнопка **ABB Library** открывает слайд-меню для выбора роботов, позиционеров и платформ различного назначения (рисунок 1.3).

Выберите из меню робота модели IRB 2600 для установки его на площадке РТК. После установки робота на площадке в окне **Output** появится сообщение о результатах импорта цифрового прототипа робота: *Imported C:\Program Files\ABB Industrial IT\Robotics IT\RobotStudio 5.13\ABB Library\Robots\IRB2600_12_165__01.rslib 5.02.2012 11:19:10.*



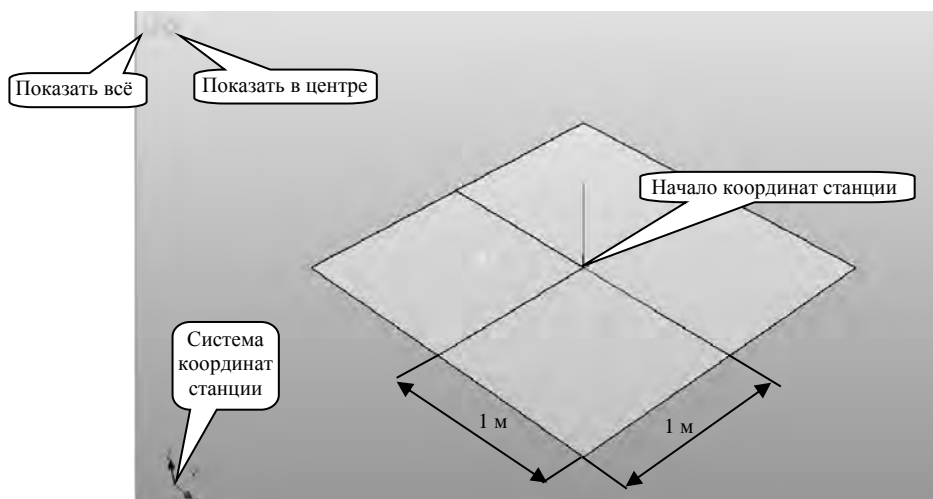


Рисунок 1.2 – Окно создания «пустой» станции



Рисунок 1.3 – Окно слайд-меню для выбора оборудования из библиотеки ABB Library

Обратите внимание на то, что виртуальный *контроллер не подключен*, в нижнем правом углу окна поле **Controller Status** имеет значение 0/0 и не окрашено в зеленый цвет.

Кнопка **Import Library** левой части меню **Home** открывает доступ для импорта цифровых прототипов, размещенных в других папках библиотеки.

Нажмите кнопку **Import Library**, а затем кнопку **Browse for Library**. Открывается окно для выбора папок других разделов библиотеки технологического, вспомогательного оборудования и инструментов. В левой части открывшегося окна щелкните название папки **ABB Library** для доступа к другим папкам библиотеки. Открывается окно для выбора папок, которые содержат файлы цифровых прототипов технологического, вспомогательного оборудования и инструментов (рисунок 1.4).

Поочередно открывая папки и выбирая файлы цифровых прототипов оборудования, ознакомьтесь с содержанием всех папок библиотеки.

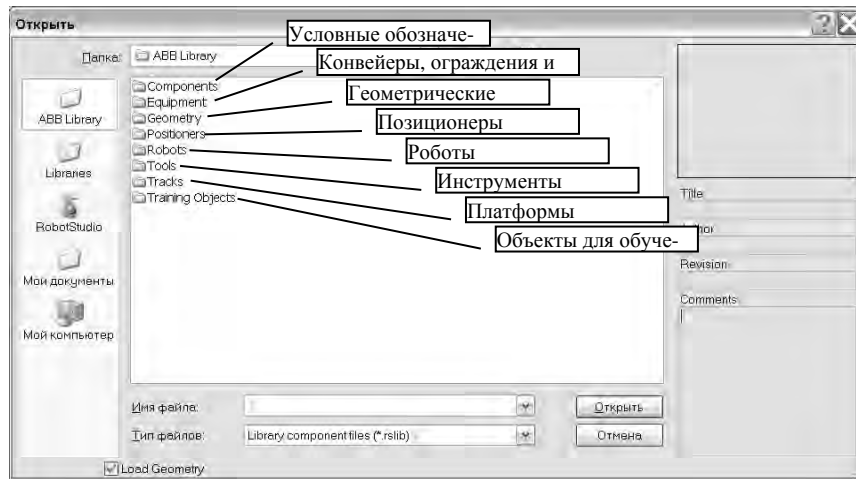


Рисунок 1.4 – Окно для выбора папок библиотеки ABB Library

Для увеличения высоты рабочей зоны робота на 240 мм из папки **Equipment** (Оборудование) выберите пьедестал **Robot Pedestal 1400 H240** и импортируйте его в моделируемый РТК.

Поднимите робота на 240 мм по оси Z для установки его на пьедестал с помощью команды **Set Position** из контекстного меню объекта-робота **IRB2600-12_165_01** на дереве проекта закладки **Layout**.

Из папки **Equipment** выберите и импортируйте **IRC5_Drive-Module** – силовой блок управления двигателями робота. Установите блок управления за пределами рабочей зоны робота. Установку цифрового прототипа блока управления в заданное место выполните, используя команды перемещения вручную из раздела **FreeHand** меню **Home**.

Из той же папки **Equipment** выберите и импортируйте прототип контроллера робота **IRC5_Control-Module** и установите его в направляющие, которые расположены на верхней панели блока управления двигателями робота. Установку в заданное место цифрового прототипа контроллера выполните с помощью команды перемещения вручную из раздела **FreeHand** меню **Home**. Для правильной оценки величины и направления перемещений используйте команды управления видами **View** и команду увеличения изображения (ролик мышки).

Для выбора оборудования (позиционера, конвейера, платформы робота или инструмента) нажмите кнопку **Import Library**, затем кнопку **Browse for Library**. В окне проводника слева (см. рисунок 1.4) щелкните название папки **ABB Library** и выберите соответствующую папку. Для вставки позиционера **Irbp250a_m2001_rev1** откройте папку **Positioners** и импортируйте заданный позиционер (библиотечный файл с расширением ***.rslib**). Разместите стол позиционера в середине рабочей зоны робота. Для этого выполните команду **Set Position** из контекстного меню объекта-позиционера **Irbp250a_m2001_rev1** на дереве проекта закладки **Layout** и введите координату **X = 1000 мм**.

Для обеспечения безопасной работы РТК из папки **Equipment** импортируйте цифровые прототипы ограждения **Fence_740** (секция шириной 750 мм), **Fence_2500** (секция шириной 2500 мм), **Fence_Gate** (секция с дверью общей шириной 1500 мм).

Перемещение в заданное место цифровых прототипов секций ограждения выполните с помощью команды перемещения вручную из раздела **FreeHand** меню **Home**. Для правильной оценки величины и направления перемещений используйте команды управления видами **View**. Для повышения точности стыковки секций ограждения увеличивайте изображение и применяйте привязку.

При установке секций ограждения необходимо рационально использовать производственные площади цеха и не допускать возможности столкновения руки робота с ограждением. Дверь разместите рядом с контроллером. Это позволит получить быстрый и безопасный доступ к контроллеру для выключения электропитания РТК при аварийной ситуации.

Инструмент должен быть установлен непосредственно на фланец руки робота точно в точке начала системы координат фланца (рисунок 1.5).

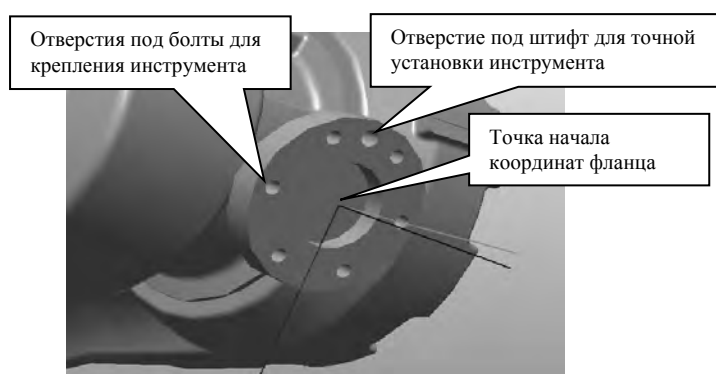


Рисунок 1.5 – Фланец руки робота для установки инструмента

Стандартные инструменты (для сварки, окраски) могут быть выбраны из библиотеки **ABB Library**. Установите на фланец робота сварочную головку для дуговой сварки в среде углекислого газа **AW_Gun_PSF_25** из папки **Tools**.

Нажмите кнопку **Import Library**, затем кнопку **Browse for Library** (просмотр библиотеки). В окне проводника слева (рисунок 1.4) щелкните название папки **ABB Library** и откройте папку **Tools** (Инструменты). Для импорта сварочной головки выберите файл **AW_Gun_PSF_25.rslib**. После нажатия кнопки **Открыть** геометрическая модель сварочной головки разместится в начале координат моделируемой станции (см. рисунок 1.2).

Для точной установки инструмента на фланец робота на вкладке **Layout** выберите узел **AW_Gun_PSF_25**. Откройте контекстное меню и выберите команду **Attach to** (присоединить к ...). В открывшемся списке нажмите пункт меню **IRB2600-12_165_01**. Появится окно с запросом к пользователю «Следует ли сохранить текущую позицию инструмента».

Нажмите кнопку **Нет**. В графическом окне видно, что сварочная головка точно установлена в точку начала системы координат фланца. Вращением, увеличением и перемещением изображения оцените точность установки инструмента на фланец.

Работа с цифровыми прототипами из библиотеки пользователя. Состав цифровых прототипов оборудования и других средств оснащения в библиотеке системы RobotStudio ограничен. Во многих случаях для моделирования работи-

зированных технологических процессов требуются оригинальные цифровые прототипы схватов, кассет, палет и др. специальной технологической оснастки.

Цифровые прототипы специальных средств технологического оснащения могут быть созданы средствами CAD-систем, например, в среде AutoCAD, SolidWorks, или других графических систем.

Созданные в CAD-системе цифровые прототипы конвертируются в формат файлов импорта системы RobotStudio ***.sat, (ACIS) *.3ds (3DStudio)** и др. и затем могут быть использованы для моделирования технологического оснащения проектируемого РТК.

Файлы оригинальных цифровых прототипов обычно размещают в папке **Geometry**, расположенной на диске C: **C:\Program Files\ABB Industrial IT\Robotics IT\RobotStudio 5.13\ABB Library\Geometry** или в папке «Мои документы» **\Мои документы\RobotStudio\ Geometry**.

Для использования оригинальных цифровых прототипов пользователя скопируйте ранее созданные файлы цифровых прототипов из папки **Лаб2_Геометрия** в папку **Geometry** по указанному выше пути.

Импортируйте оригинальные цифровые прототипы. Нажмите верхнюю часть кнопки **Import Geometry**. Появится окно проводника **Открыть**. В окне папки **Geometry** выберите файл **СтолP** и откройте его нажатием кнопки **Открыть**.

Для придания столу правильного положения с помощью команды **Set Position** поверните его на 90 град вокруг оси X, переместите на 500 мм по оси Z и X, переместите на –550 мм по оси Y. Аналогичным образом вставьте объект **Груз**.

Для правильного размещения груза на плоскости стола выполните измерения размеров груза, стола и расстояния от груза до стола, используя измерительные инструменты раздела меню **Measure** (Измерения) (рисунок 1.6) закладки **Modeling** главного меню.

Для выполнения измерений установите вид сверху – **Top**. Последовательно нажмите кнопки **Point tu Point** (измерение расстояния между указанными точками) и кнопку объектной привязки к конечной точке отрезка для повышения точности измерений. С помощью указателя задайте точки начала и конца измерения для определения размеров груза, а затем и поверхности стола.

Для измерения расстояния между объектами щелкните объект мышкой для выделения его границ. Указателем отметьте на границе точку начала измерения. Выделите второй объект и укажите конечную точку измерения. Между точками начала и конца измерения появится надпись с указанием расстояния между объектами.

После выполнения измерений определите, на какое расстояние необходимо переместить груз по осям **X, Y, Z** с помощью команды **Set Position**.

Подключение виртуального контроллера. Для синхронизации контроллера и робота откройте в главном меню закладку **Offline**. Щелкните кнопку **Synchronize**, а затем **Synchronize to Station**.

В открывшемся диалоговом окне на дереве проекта выберите название модели робота, к которому будет подключен виртуальный контроллер.

Индикатор состояния виртуального контроллера в правом нижнем углу ок-



на RobotStudio становится зеленым и появляется запись о состоянии виртуального контроллера **Controller status: 1/1**. В окне **Output** будет выведено сообщение: **Synchronization to Station completed** (Синхронизация станции завершена).

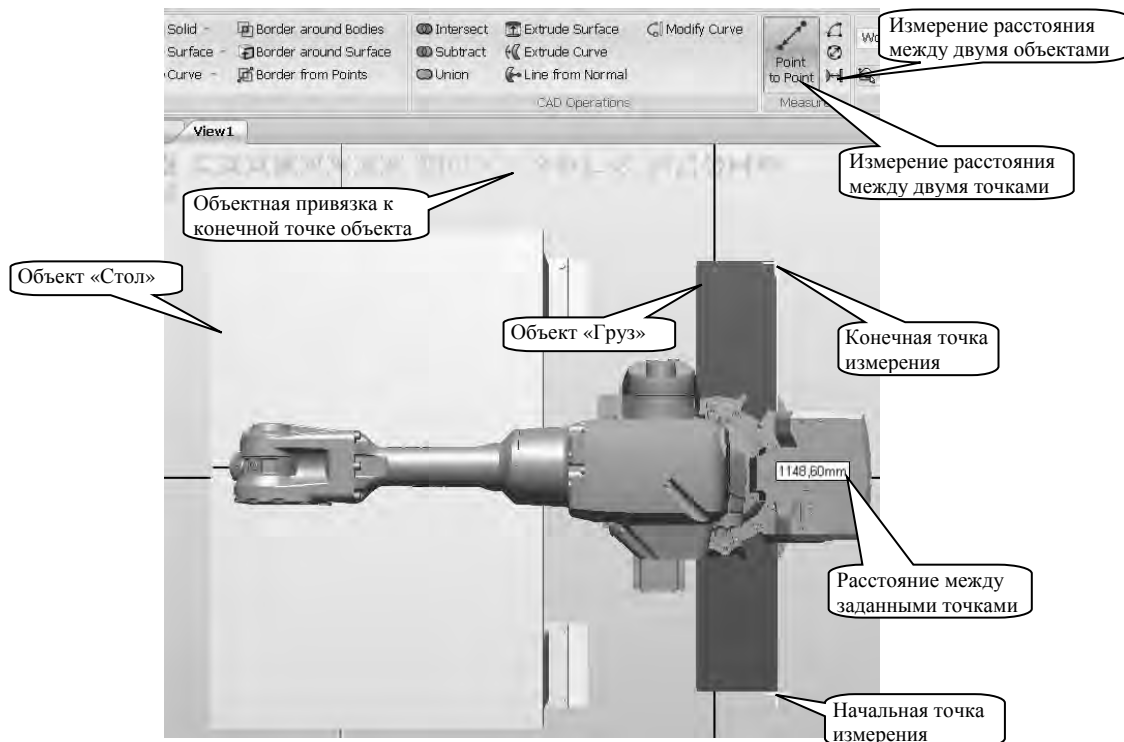


Рисунок 1.6 – Измерение размеров объектов и расстояний между ними

Индивидуальное задание.

Вариант 1: Создать РТК на базе робота IRB 2400L, состоящего из блока управления роботом IRC5_Dual-Cabinet, краскопульта ECCO_70AS__03, конвейера модели Conveyor 1320 4800 h2B, груза B1 и ограждения. В ограждении предусмотреть проходы для загрузки деталей на конвейер и их выгрузки с конвейера.

Вариант 2: Создать РТК на базе робота IRB 4400/60, состоящего из блока управления роботом IRC5_Dual-Cabinet, сварочной головки PKI_500_di_M2001, конвейера модели Conveyor 2400 3str h2 груза B2 и ограждения. В ограждении предусмотреть проходы для загрузки деталей на конвейер и их выгрузки с конвейера.

1.3 Определение движений элементов технологического оборудования и оснастки

Следует создать РТК (станцию) на базе шаблона IRB 660_250kg_3.15m.

Для использования оригинальных цифровых прототипов скопируйте файлы цифровых прототипов оборудования из папки **Лаб3 Геометрия** в папку **Geometry** по следующему пути: **C:\Program Files\ABB Industrial IT\Robotics IT\RobotStudio 5.13\ABB Library\Geometry**.

Из библиотеки цифровых прототипов **Geometry** на площадку РТК установите конвейеры **conv_box_top**, **conv_pallets_top**.

Установите силовой блок и контроллер IRC5 в пространстве станции на безопасном расстоянии от робота.

Из библиотеки цифровых прототипов **Geometry** на площадку РТК установите **Стол** (это пьедестал для робота). Командой **Set Position** определите положение стола с координатами $X = 450$, $Y = 350$, $Z = 540$, углы поворота вокруг осей X и Y – 90 град.

Переместите робота и его систему координат на пьедестал. Командой **System Configuration** (Конфигурирование системы) на вкладке **Offline** задайте новое положение робота и его системы координат, как показано на рисунке 1.7.

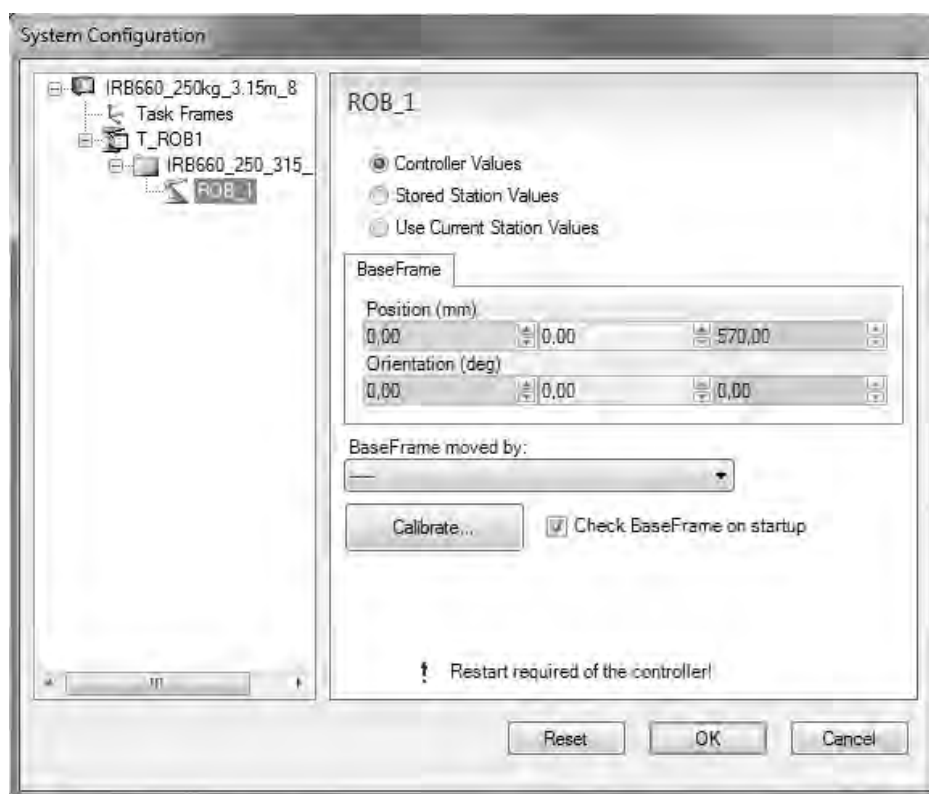


Рисунок 1.7 – Перемещение робота и его системы координат

В процессе изменения конфигурации появится запрос на рестарт контроллера. Необходимо подтвердить перезагрузку контроллера нажатием кнопки **Да**.

Создайте объект **Груз**. На вкладке **Modeling** в меню **Solid** выберите объект **Box** (ящик) и задайте его габаритные размеры:

Length (длина): 210 мм;

Width (ширина): 1000 мм;

Height (высота): 200 мм.

Командой **Set Position** определите положение груза, задавая координаты: $X = 2000$, $Y = 110$, $Z = 1400$, угол поворота вокруг оси X – 180 град.

Из библиотеки цифровых прототипов **Geometry** на площадку РТК установите палету (файл **Pallet.sat**). Командой **Set Position** задайте координаты ее положения $X = 1700$, $Y = -350$, $Z = 415$, угол поворота вокруг оси Y – -90 град.

Создание механизма открытия-закрытия схвата. Загрузите файлы цифровых прототипов деталей схвата **tool_part1** и **tool_part2** из библиотеки **Geometry**. Переместите **tool_part2**, задавая командой **Set Position** его координаты: $X = 0$, $Y = 180$, $Z = 55$, угол поворота вокруг осей – 0 град.

В верхней части окна программы выберите вкладку **Modeling** и нажмите кнопку **Create Mechanism** (Создать механизм).

В окне **Create Mechanism** (рисунок 1.8) введите название механизма **My_Gripper** (мой схват) для его идентификации на дереве проекта. Выберите тип механизма **Tool** (инструмент).

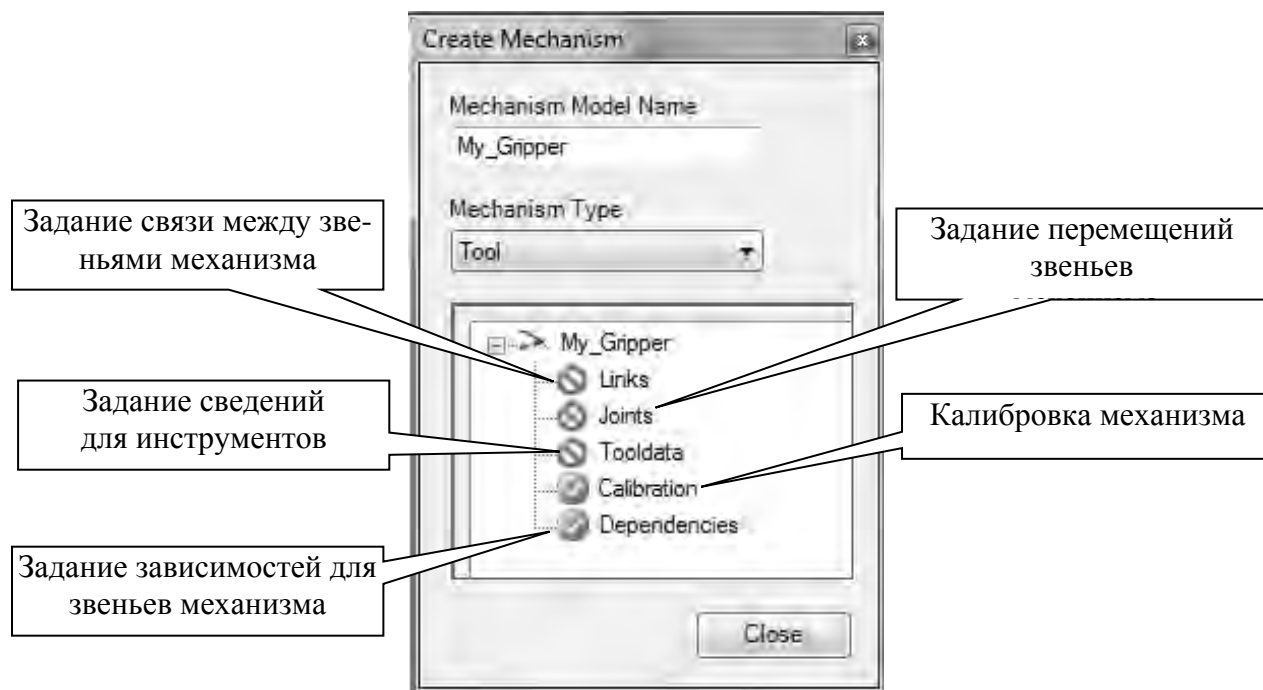


Рисунок 1.8 – Окно Create Mechanism

Выделите узел **Links** (Связи) и вызовите его контекстное меню нажатием правой клавиши мыши. В контекстном меню объекта **Links** щелкните **Add link** (Добавить связь). Открывается окно **Create Link** для создания связей между подвижными элементами схвата.

В диалоговом окне **Create Link** (Создайте Связь) в поле **Selected Part** (Выберите звено) выберите **tool_part1** (главная неподвижная часть схвата), поставьте галочку напротив **Set as BaseLink** (Установить как основную связь).

Для выбора связываемого звена схвата нажмите кнопку с зеленой стрелкой. Название связанной части (звена) **tool_part1** появится в поле списка справа. Нажмите кнопку **Apply** (Применить). В окне появится поле **Link Name** (имя связи) со значением **L1**. В поле **Selected Part** (Выбрать звено) из списка выберите **tool_part2**, нажмите на кнопку с зеленой стрелкой. Звено **tool_part2** появится в правом поле.

В результате выполнения этих операций созданы связи между подвижными частями схвата. При необходимости положения звеньев могут быть изменены путем задания значений координат и углов поворота в полях **Part Po-**

sition (Положение звена) и **Part Orientation** (Ориентация звена, части). Нажмите кнопку **OK**.

Задание перемещений подвижных элементов схвата. В окне **Create Mechanism** (см. рисунок 1.8) выделите узел **Joints** (Звенья) и вызовите его контекстное меню нажатием правой клавиши мыши. В контекстном меню объекта **Joints** щелкните **Add Joint** (Добавить звено). Появится новое окно **Create Joint** (Создание звена) (рисунок 1.9).

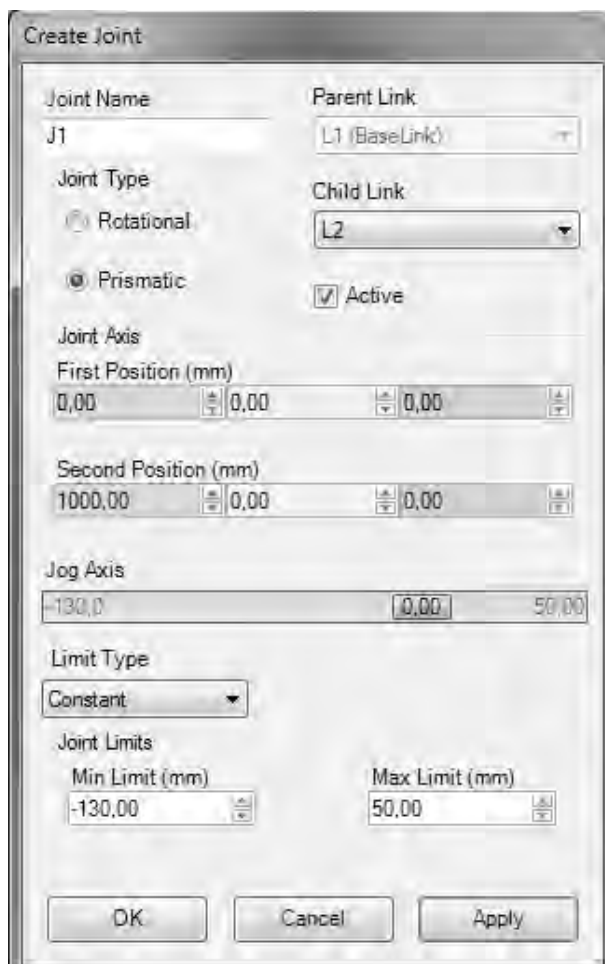


Рисунок 1.9 – Окно Create Joint

Назначение параметров, задаваемых в окне **Create Joint**, приведено в таблице 1.2.

Задайте значения параметров так, как это показано на рисунке 1.9.

В окне **Create Mechanism** (см. рисунок 1.8) выделите узел **Tooldata** (Данные инструмента) (в данном случае схвата) и вызовите его контекстное меню нажатием правой клавиши мыши. В контекстном меню объекта **Tooldata** щелкните **Add Tooldata** (Добавить данные инструмента). Появится новое окно **Create Tooldata** (Создание звена).

Назначение параметров, задаваемых в окне **Tooldata**, приведено в таблице 1.3.

Назначение массоинерционных характеристик инструмента необходимо для выбора скоростей перемещений, при которых не будет создаваться опасных

суммарных гравитационных и инерционных нагрузок на механизмы привода отдельных звеньев руки робота.

Таблица 1.2 – Назначение параметров, задаваемых в окне Create Joint

Параметр	Назначение параметра
Joint Name	Определяет имя звена схвата. Имя предлагается системой
Joint Type.	Определяет тип перемещения звена схвата Rotational – вращательное, Prismatic – линейное. Выбор по умолчанию является Вращательным . Изменение типа очищает данные, расположенные в полях ниже
Parent Link	Определяет родительскую связь деталей механизма, обычно родителем является первый сустав механизма
Child Link	Определяет связь с дочерней частью механизма
Active	Эта опция делает звено активным. Активное звено – то, которое может переместить пользователь, в то время как бездействующее звено является по отношению к нему подчиненным
Joint Axis	Эта группа полей служит для определения начальной и конечной точек оси, вокруг которой вращается дочернее звено или отрезок, вдоль которого дочернее звено перемещается
First Position	Определяет точку начала вектора оси
Second Position	Определяет конечную точку вектора оси
Jog Axis	Определяет, каким образом расположена дочерняя связь относительно оси
Limit Type	Определяет тип пределов перемещения звеньев в каждом направлении, в котором может двигаться звено. Варианты: Constant – Постоянные , Variable – Переменные и No – Нет
Joint Limits	Эта группа полей для задания численных значений пределов перемещения звена
Min Limit	Определяет минимальное перемещение звена
Max Limit	Определяет максимальное перемещение звена

Таблица 1.3 – Назначение параметров, задаваемых в окне Tooldata

Параметр	Назначение параметра
Frame/Tool Data name	Определяет название группы данных инструмента
Belongs to Link	Определяет связь, к которой принадлежит группа данных
Position	Определяет новое положение инструмента в пространстве
Orientation	Определяет новую ориентацию инструмента в пространстве
Select values from target/Frame	Флаг устанавливается для определения данных из поля, расположенного ниже
Tool Data	Группа данных для задания массоинерционных характеристик схвата или другого инструмента. Эта группа данных видима, если механизм – инструмент
Mass	Определяет массу инструмента
Center of Gravity	Определяет положение центра тяжести инструмента
Moment of Inertia Ix, Iy, Iz	Определяет моменты инерции инструмента относительно осей

Введите данные в соответствии с рисунком 1.10. Окно **Create Mechanism** приобретает вид, показанный на рисунке 1.11.

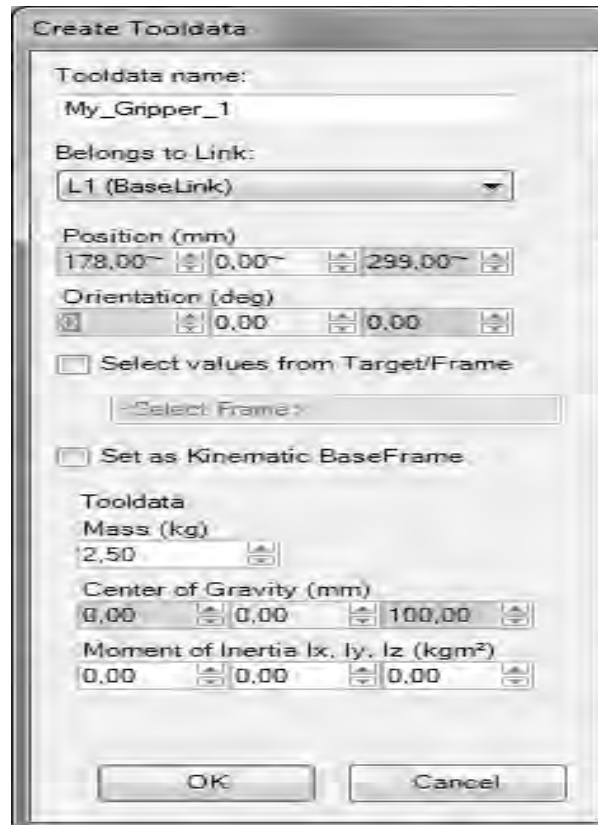


Рисунок 1.10 – Окно Create ToolData

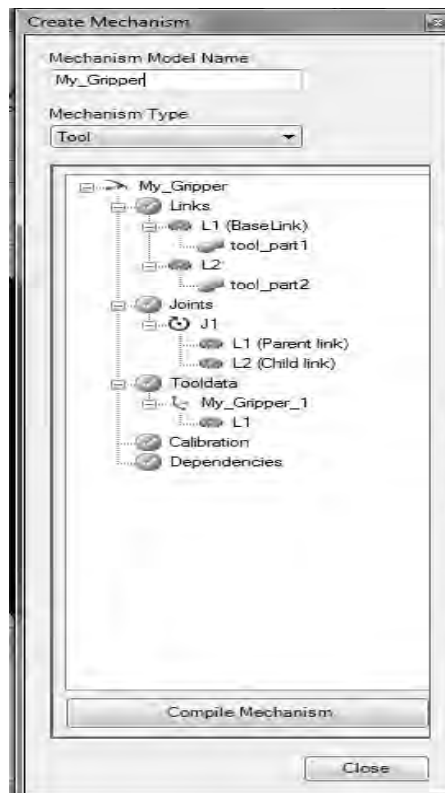


Рисунок 1.11 – Вид окна Create Mechanism после ввода данных об инструменте

Нажмите кнопку **Compile Mechanism** (Сборка Механизма) в нижней части окна **Create Mechanism** (см. рисунок 1.11).

Под окном **Poses** (поза, положение) нажмите кнопку **Add**. В появившемся окне **Modify Pose** (изменение позы) для задания положения схвата в состоянии «Закрит» введите данные, как на рисунке 1.12.

Задайте положение схвата для состояния «Открыт» (рисунок 1.13).

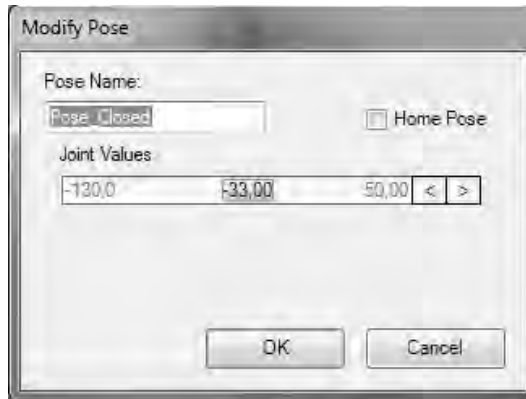


Рисунок 1.12 – Создание позы схвата в положении «Закрит»

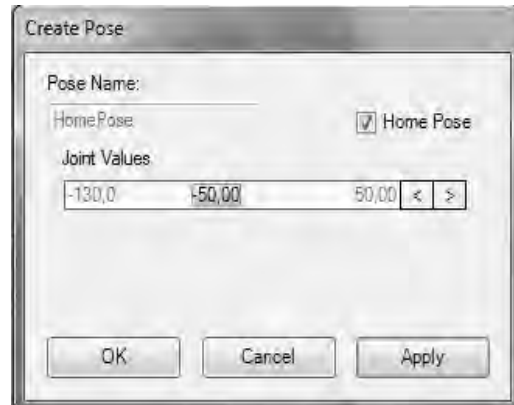


Рисунок 1.13 – Создание позы схвата в положении «Открыт»

На следующем этапе моделирования работы механизма схвата задается время, которое необходимо на его открытие и закрытие. Нажмите кнопку **Set Transmission Time** (Задать время перемещения) (рисунок 1.14).

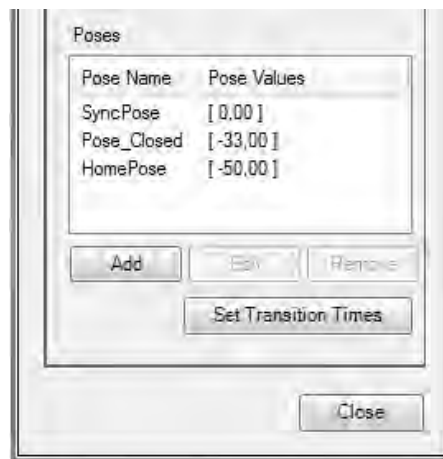


Рисунок 1.14 – Кнопка Set Transmission Time

В окне **Set Transmission Time** (рисунок 1.15) отображаются имена ранее определенных положений звеньев механизма. В столбце **To Pose** указываются стартовые положения механизма, в строках **From Pose** – конечные положения. Для каждой комбинации исходного и конечного положения механизма может быть задано время срабатывания механизма (в секундах) при переходе его из одного положения в другое. Введите данные, показанные на рисунке 1.15.

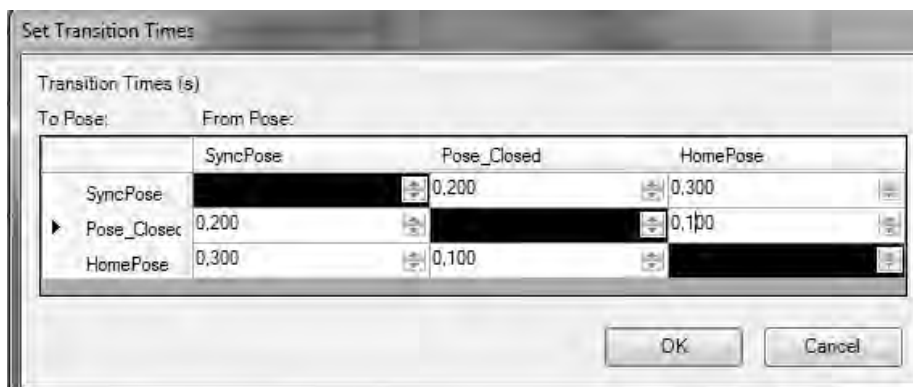


Рисунок 1.15 – Окно Set Transmission Time

После ввода данных, которые определяют время срабатывания механизма, создание механизма считается завершенным, и система выводит сообщение с запросом на подтверждение завершения процедуры моделирования механизма.

После завершения процедуры моделирования работы механизма схвата на дереве проекта в закладке **Layout** появится ветвь **My_Gripper**.

Установка схвата на фланец руки робота. Для точной установки схвата на фланец робота на вкладке **Layout** выберите ветвь **My_Gripper**. Откройте контекстное меню и выберите команду **Attach to** (присоединить к ...). В открывшемся списке нажмите пункт меню **IRB 660_250_315_01**. Появится окно с запросом к пользователю «Следует ли сохранить текущую позицию инструмента». Нажмите кнопку **Нет**. В графическом окне видно, что схват установлен в точку начала системы координат фланца. Вращением, увеличением и перемещением изображения оцените точность установки инструмента на фланец.

Файл с результатами моделирования РТК и механизма схвата необходимо сохранить в своей рабочей папке. Результаты моделирования будут использованы при выполнении практической работы № 2.

Индивидуальное задание.

Создать РТК на базе шаблона по заданному варианту (таблица 1.4).

Таблица 1.4 – Варианты индивидуальных заданий

Номер варианта	Модель робота	Модель конвейера	Длина груза	Ширина груза	Высота груза
1	IRB 4400	950 4000 h2	150	800	150
2	IRB 2600	1200 4000 h2	250	1200	180

Установите силовой блок и контроллер IRC5 в пространстве станции на безопасном расстоянии от робота.

Из папки **Equipment** выберите и импортируйте конвейер по заданному варианту. Командой **Set position** разверните конвейер на -90 град и перенесите на 1000 мм по оси X.

Из папки **Equipment** выберите и импортируйте палету **Euro Pallet**.

Командой **Set position** разверните и перенесите палету таким образом, чтобы при перегрузке груза с конвейера на палету робот совершал минимальные перемещения в пределах своей рабочей зоны и не мешал подъезду погрузчика для забора палеты. Возможное расположение робота, конвейера, палеты и направление движения погрузчика показаны на рисунке 1.16.

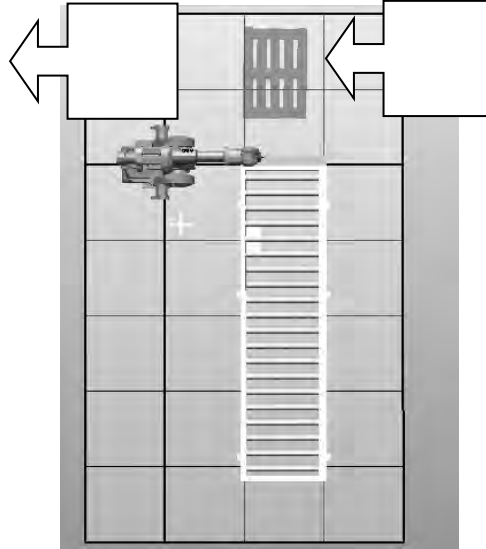


Рисунок 1.16 – Вариант компоновки РТК

Установите ограждение для создания зоны безопасности робота и конвейера. Если необходимо, измените положение контроллера и силового блока.

Создайте груз с размерами по заданному варианту и установите его на крайней позиции конвейера.

Импортировать детали схвата из библиотеки Geometry. С учетом размеров груза задайте необходимые параметры перемещения звеньев механизма схвата.

Установите схват на фланец руки робота. Оцените получившийся результат и при необходимости выполните корректировку заданных параметров звеньев механизма схвата.

Файл с результатами моделирования РТК и механизма схвата по индивидуальному заданию сохраните в своей рабочей папке.

2 Практическая работа № 2. Разработка структуры управления и компоновки ГПС (ГАЛ, ГАУ) для технологических процессов обработки деталей

2.1 Задание пути перемещения и моделирование движения рабочих инструментов робота

Следует загрузить систему **RobotStudio**, а затем файл модели РТК, который был создан при выполнении п. 1.3 практической работы № 1 (рисунок 2.1).

Создадим специальные системы координат, относительно которых будут заданы целевые точки. В **RobotStudio** такая система координат называется **WorkObject** (Объект работы).

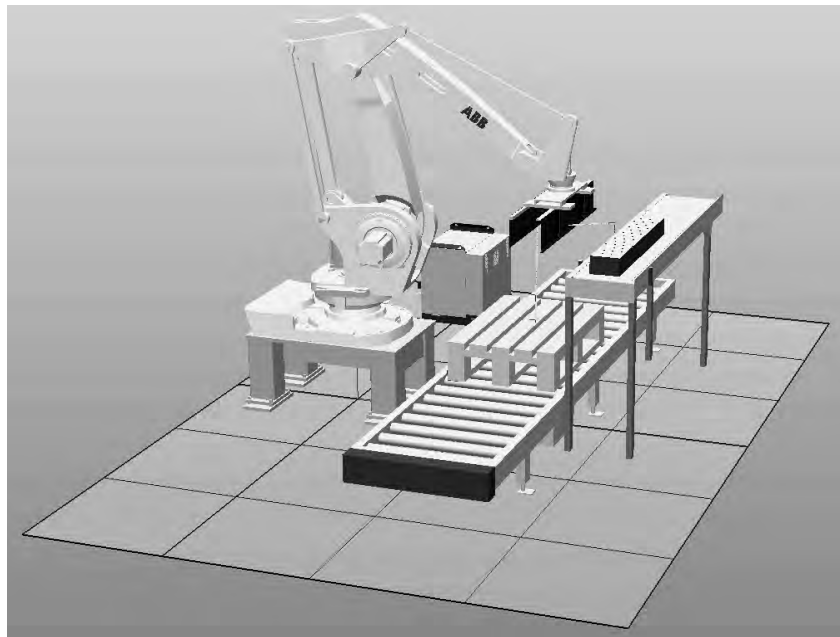


Рисунок 2.1 – Модель РТК, используемая для программирования перемещений груза

Для создания объекта работы на вкладке **Home** в подразделе **Path Programming** нажимаем кнопки **Other** → **Create WorkObject**.

В появившемся окне **Create WorkObject** (Создание объекта работы) в поле **Name** (Имя) присваиваем имя **Wobj_box** создаваемому объекту работы (рисунок 2.2).

Повернем станцию в графическом окне таким образом, чтобы ее конвейер был ориентирован так, как показано на рисунке 2.3.

Для точного определения системы координат объекта работы включаем привязку к конечным точкам. Для этого нажимаем кнопку **Snap End** верхней части графического окна системы.

Далее в разделе **User Frame** (Пользовательская Структура) в окне **Create Workobject** выбираем пункт **Frame by points** (Структура из точек) и нажимаем на поле справа от него. Появится окно ввода координат точек (рисунок 2.4).

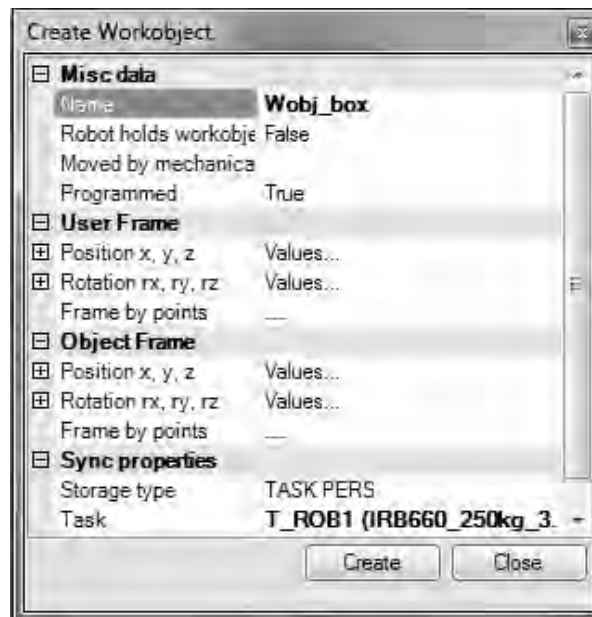


Рисунок 2.2 – Окно создания объекта работы

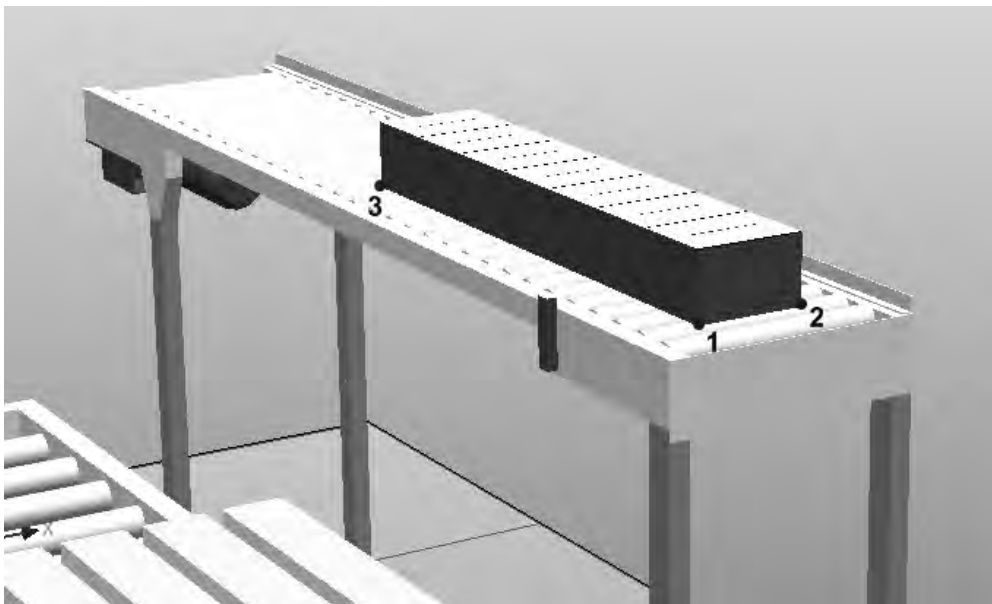


Рисунок 2.3 – Вид станции для создания Workobject

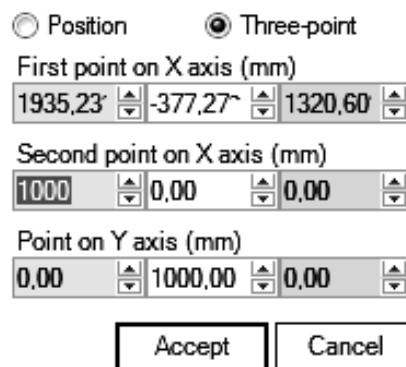


Рисунок 2.4 – Окно для ввода координат точек при создании Workobject

Выбираем опцию **Three-point** (По трем точкам). Затем щелкаем поле ввода координат первой точки, с помощью привязки наводим курсор на точку 1 (см. рисунок 2.3) и щелкаем по ней. В полях, соответствующих первой точке, появятся значения координат. Аналогично вводим координаты для второй и третьей точек (см. рисунок 2.3). Нажимаем кнопку **Accept** (Принять), в окне **Create WorkObject** – кнопку **Create** (Создать).

Аналогично создаем **WorkObject** с названием **Wobj_pallet** на основе точек, показанных на рисунке 2.5.

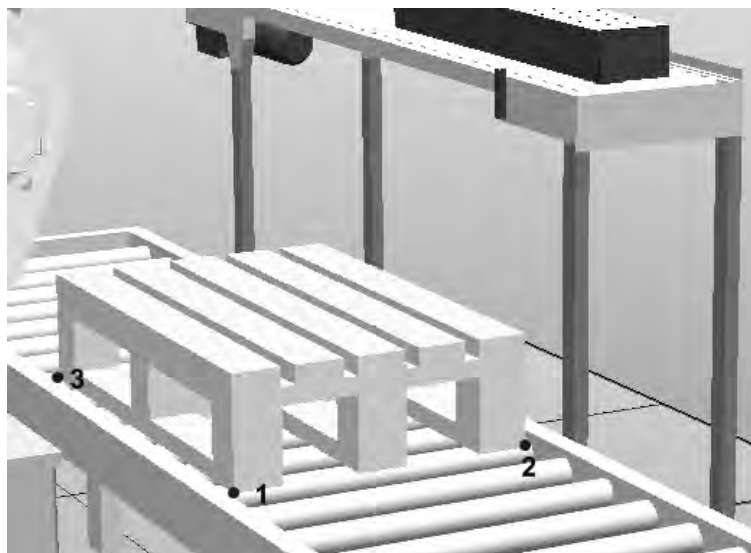


Рисунок 2.5 – Вид станции для задания Workobject с названием Wobj_pallet

Создание целевых точек для создания пути инструмента робота. Для создания пути инструмента необходимо в рабочем пространстве задать точки, которые в дальнейшем будут определять направления перемещения инструмента. На рисунке 2.6 показаны путь и целевые точки перемещения груза с конвейера на палету. Рассмотрим методику определения целевых точек путем задания их координат X , Y , Z .

Для задания точек координатами по осям X , Y , Z откройте вкладку **Home** главного меню системы в разделе **Path Programming** и нажмите кнопки **Target** → **Create Target** (Цель – целевая точка).

Появится окно ввода данных (рисунок 2.7). Внизу окна выбираем рабочий объект (WorkObject) **Wobj_box**. Заполняем поля **Position** значениями координат целевой точки **1** $X = 1930$, $Y = 210$, $Z = 1330$. Задаем в поле **Orientation** угол ориентации инструмента в этой точке относительно оси Y , равный 180 град. Нажимаем кнопку **Add** (Добавить), а затем кнопку **Create** (Создать).

Создадим целевую точку **2** над рабочим объектом **Wobj_pallet** с координатами $X = 1200$, $Y = 200$, $Z = 730$ и углом ориентации инструмента в этой точке относительно оси Y , равным 180 град. Закроем окно **Create Target** нажатием кнопки **Close**. Откроем меню **Paths&Targets** (Пути и цели) (рисунок 2.8).

Точку **Target_10**, принадлежащую **Wobj_box**, используя команду контекстного меню **Rename**, переименуем в **take_box** (взять ящик), а точку **Target_20**, принадлежащую **Wobj_pallet**, – в **put_box** (положить ящик).

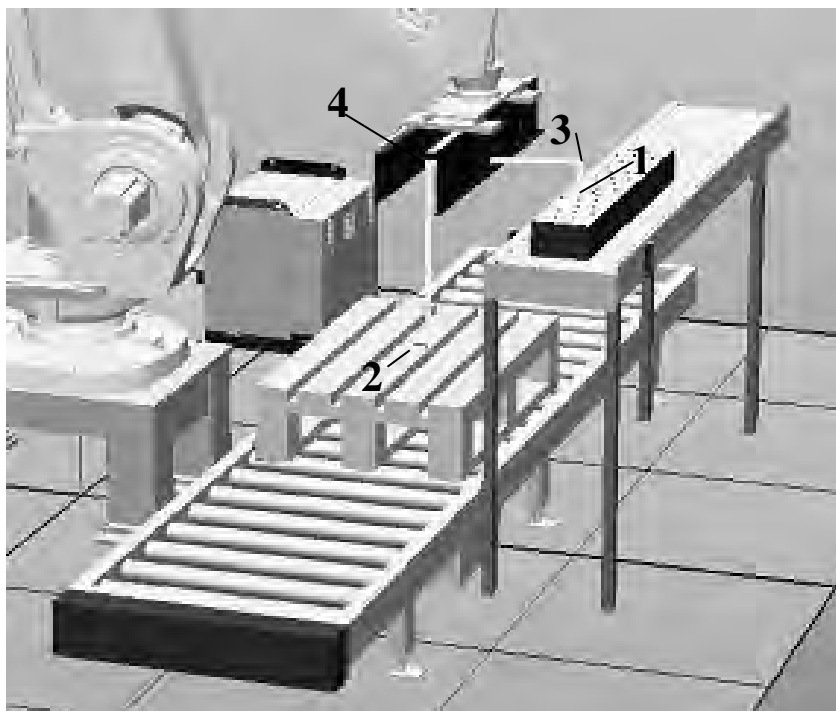


Рисунок 2.6 – Целевые точки и путь инструмента при перемещении груза на палету

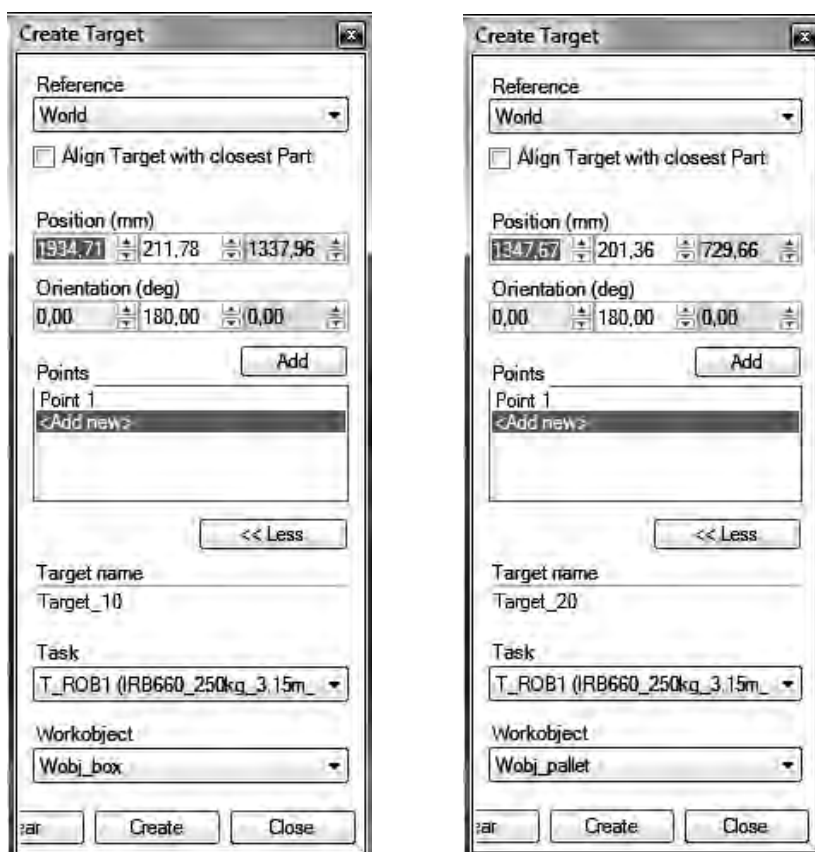


Рисунок 2.7 – Окна задания координат целевых точек 1 и 2

Выберем точку **put_box** в меню **Paths&Targets**. В контекстном меню щелкнем **Jump To Target** (Перейти к точке назначения). Появится окно конфигурирования звеньев руки робота (задание углов поворота звеньев) в выбранной

точке (рисунок 2.9). Выбираем **Cfg 1 (0,0,0,0)** и нажимаем кнопку **Apply**. В графическом окне видим, что манипулятор переместился в точку **put_box**. Повторим эти команды для точки **take_box**.

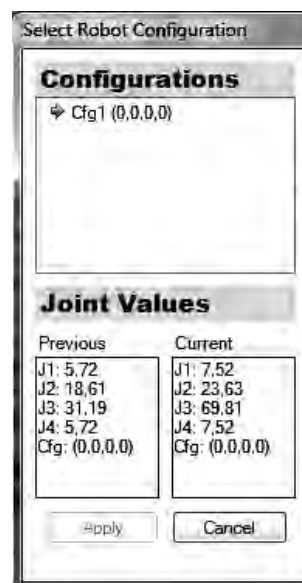
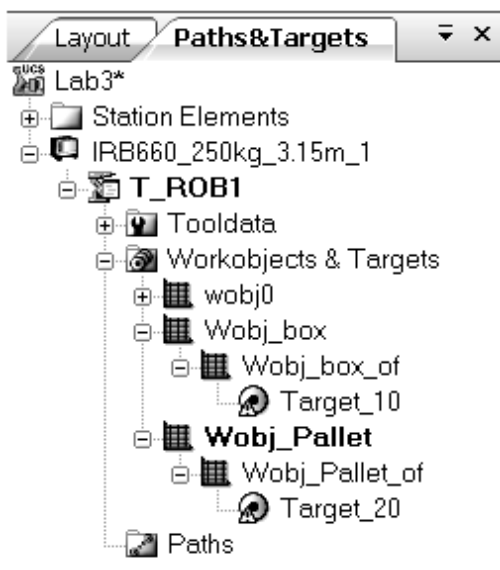


Рисунок 2.8 – Дерево путей и точек перемещения инструмента

Рисунок 2.9 – Окно задания конфигурации звеньев руки робота

При необходимости координаты целевых точек могут быть изменены командой **Modify Target** и проверены на корректность задания путем выполнения команды **Jump To Target**.

Создадим точку **3** путем копирования точки **take_box**. Выделим на дереве путей точку **take_box** и из контекстного меню выполним команду **Copy**. Выделим на дереве путей и точек ветвь **Wobj_box_of**, в контекстном меню выберем пункт **Paste** (Вставить). Появится новая точка с названием **take_box_2**. Переименуем ее в **take_box_offset** (Поднять ящик). Выделим точку **take_box_offset** и в контекстном меню выберем пункт **Modify Target** → **Set Position**. В появившемся окне (рисунок 2.10) вводим координаты точки **3** $X = 0$, $Y = 0$, $Z = -200$, которые определяют высоту подъема груза над конвейером, и нажимаем кнопку **Apply**. Аналогичным образом создадим точку **4** **put_box_offset** (Положить ящик).

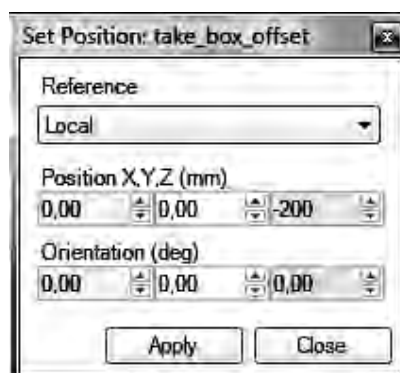


Рисунок 2.10 – Окно задания координат точки 3

Далее добавим промежуточную точку между позициями **put_box_offset** и **take_box_offset** для того, чтобы при возврате в точку **1** по кратчайшему пути манипулятор не сталкивался с конвейером и грузом. В этой же точке манипулятор будет останавливаться в конце траектории в ожидании поступления на конвейер следующего груза.

На вкладке **Home** в разделе **Settings** поменяем систему координат рабочих объектов **Workobject** на систему координат робота **wobj0**. Выведем руку робота в нужную точку вручную с помощью команды **Jog Linear** из раздела меню **Free-hand**. Затем из раздела меню **Path Programming** (Программирование Пути) нажмем кнопку **Target** (Цель) и выберем пункт **Teach Target** (Передать Цель). Переименуем на дереве путей и целей полученную точку в **Home_position**.

Создание пути перемещения груза. Для создания пути инструмента по ранее определенным точкам нажимаем в разделе меню **Path Programming** кнопки **Path** → **Empty Path** (Пустой Путь).

На дереве путей и точек **Paths&Targets** в разделе **Path** появится пустой путь **Path_10**. Добавим в пустой путь **Path_10** целевые точки. Для этого выделяем в меню **Paths&Targets** целевую точку **Home_position**. В контекстном меню выбранной точки нажимаем пункт меню **Add to path** → **Path 10** → **<First>**. На дереве путей и точек **Paths&Targets** в разделе **Path_10** появится участок пути к заданной точке **MoveL Home_position**. Создадим полный путь командой **Add to path** → **Path 10** → **<Last>**, добавляя другие точки в последовательности, показанной на рисунке 2.11.

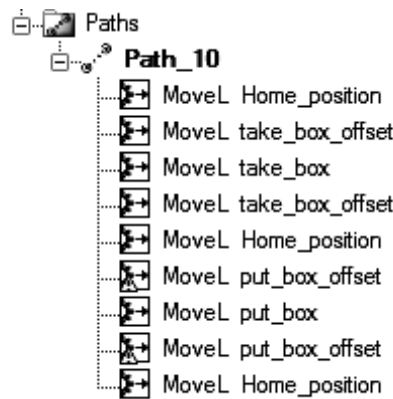


Рисунок 2.11 – Полный путь инструмента, заданный отрезками прямых MoveL

На дереве проекта выбираем путь **Path_10** и из контекстного меню выполняем команду **Auto Configuration**.

Для проверки созданного пути инструмента выбираем вкладку **Simulation** (Моделирование), раздел меню **Configure** (Формирование) и нажимаем пункт меню **Simulation Setup** (Установки Моделирования).

Появится окно **Setup Simulation** (рисунок 2.12). В списке **Available Procedures** (Доступные Процедуры) выделяем путь **Path_10** и нажимаем на стрелку влево, а затем кнопки **Apply** и **OK**.

В результате выполнения этих команд будет сформирована программа, управляющая движением робота вдоль заданной траектории.

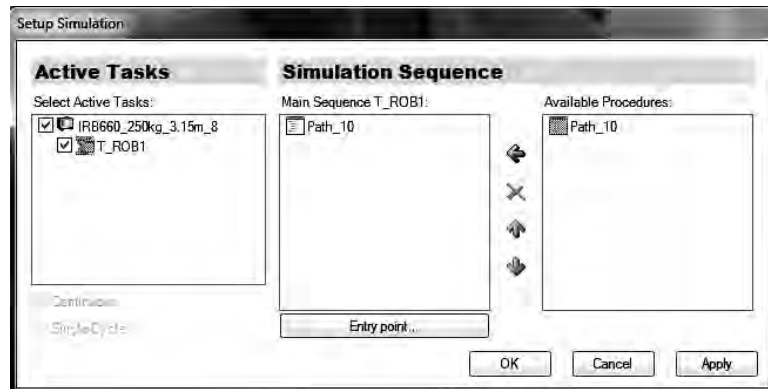


Рисунок 2.12 – Окно настройки моделирования Setup Simulation

Для моделирования движения робота в меню **Simulation Control** (Контроль Моделирования) нажимаем кнопку **Play**. В графическом окне системы наблюдаем, как робот перемещается по заданному пути.

Оптимизация пути перемещения груза. Созданный путь состоит из отрезков, расположенных под прямыми углами. Путь, созданный таким способом, не является оптимальным. Для того чтобы путь был оптимальным, перемещение должно происходить при минимальных движениях отдельных звеньев руки робота.

Оптимизируем путь перемещения груза путем редактирования управляющей программы, автоматически составленной системой на языке **Rapid**. Язык **Rapid** разработан исследовательским центром робототехники **ABB Group** для программирования роботов собственного производства.

Для открытия автоматически созданной программы выбираем вкладку **Offline**. В окне **Offline** на дереве программных модулей выделяем ветвь **Module1** (рисунок 2.13) и в меню **Controller Tools** (Инструментарий контроллера) нажимаем кнопку **Rapid Editor** (Текстовый редактор Rapid). Если в левой части окна закладка не появляется, то необходимо выполнить команды настройки быстрого доступа к меню **Offline Windows Layout** → **Default Layout** или **Windows Layout** → **Windows** → **Offline** (см. рисунок 2.13).

Произведем изменения в тексте программного модуля **PROC Path_10()**. В строке программного модуля **Path_10**, которая определяет начало перемещения

MoveL Home_position,v1000,z100,My_Gripper_1\WObj:=wobj0;

и в подобной строке в конце программного модуля **Path_10**, которая определяет конец перемещения

MoveL Home_position,v1000,z100,My_Gripper_1\WObj:=wobj0;

заменяем команды **MoveL** на **MoveJ**.

Это означает, что манипулятор будет перемещаться в эти точки таким образом, чтобы движение звеньев его руки было минимальным.

В строках программного модуля **Path_10**, которые определяют перемещение в точки **take_box**

MoveL take_box,v1000,z100,My_Gripper_1\WObj:=Wobj_box;

и *put_box*

MoveL put_box,v1000,z100,My_Gripper_1\WObj:=Wobj_pallet;

заменяем **z100** на **fine** (точно). Это значит, что робот должен переместить груз в эти точки точно (рисунок 2.14).

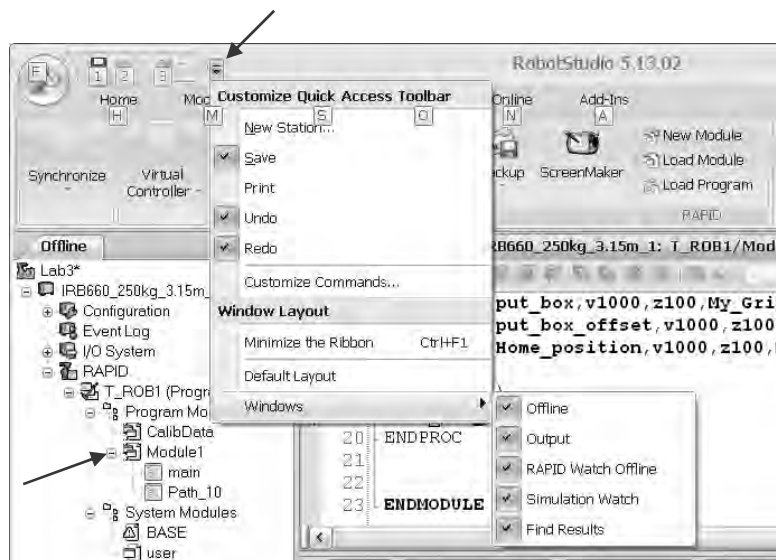


Рисунок 2.13 – Окно дерева программных модулей Offline

```

1  MODULE Module1
2      CONST robtarget Home_position:=[[350,560,915],[6.1230317691118
3      CONST robtarget take_box_offset:=[[0,710,207.04963994424],[6.1
4      CONST robtarget take_box:=[[0,710,7.04963994423968],[6.1230317
5      CONST robtarget put_box_offset:=[[350,560,515],[6.123031769111
6      CONST robtarget put_box:=[[350,560,315],[6.12303176911189E-17,
7  PROC Path_10 ()
8      MoveJ Home_position,v1000,z100,My_Gripper_1\WObj:=Wobj_Pallet;
9      MoveL take_box_offset,v1000,z100,My_Gripper_1\WObj:=Wobj_box;
10     MoveL take_box,v1000,fine,My_Gripper_1\WObj:=Wobj_box;
11     MoveL take_box_offset,v1000,z100,My_Gripper_1\WObj:=Wobj_box;
12     MoveL Home_position,v1000,z100,My_Gripper_1\WObj:=Wobj_Pallet;
13     MoveL put_box_offset,v1000,z100,My_Gripper_1\WObj:=Wobj_Pallet
14     MoveL put_box,v1000,fine,My_Gripper_1\WObj:=Wobj_Pallet;
15     MoveL put_box_offset,v1000,z100,My_Gripper_1\WObj:=Wobj_Pallet
16     MoveJ Home_position,v1000,z100,My_Gripper_1\WObj:=Wobj_Pallet;
17 -ENDPROC
18 PROC main ()
19     Path_10;
20 -ENDPROC
21
22
23 -ENDMODULE

```

Рисунок 2.14 – Окно текстового редактора системы программирования RAPID

Строка программы для задания перемещения объекта содержит данные, расшифрованные на рисунке 2.15.

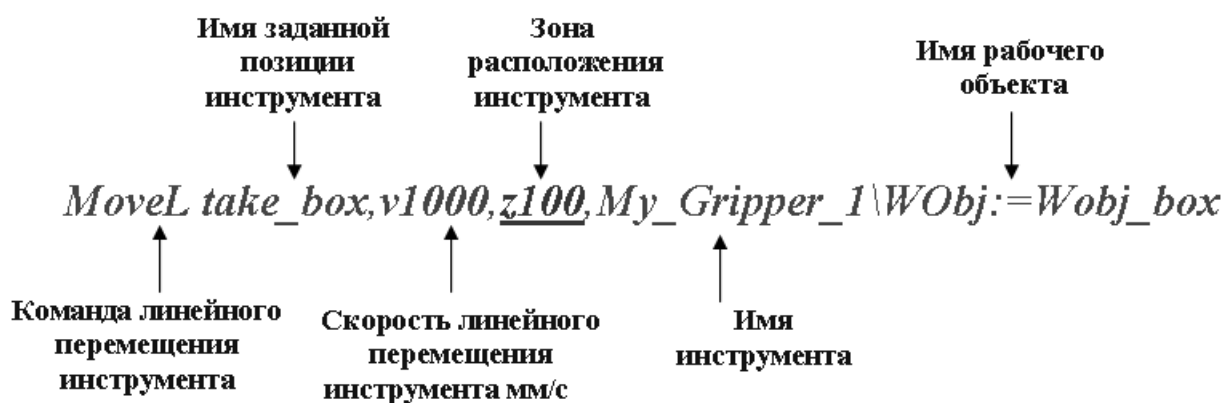


Рисунок 2.15 – Строка программы для задания перемещения объекта

Нажмем кнопку **Apply Changes** (Принять изменения), которая расположена на панели над текстом программы. Внизу в окне **Output** должно появиться сообщение об отсутствии ошибок. Закрываем окно **Rapid Editor**.

Для моделирования движения робота по измененной программе в меню **Simulation Control** (Контроль Моделирования) нажимаем кнопку **Play**. В графическом окне системы наблюдаем, как робот перемещается по оптимальному пути.

Индивидуальное задание.

Создать путь перемещения груза в РТК, который был разработан при выполнении индивидуального задания в п. 1.3 практической работы № 1.

Осуществить оптимизацию пути редактированием программного модуля.

В ходе построения модели РТК следует вести протокол. Результаты моделирования будут использованы при выполнении индивидуального задания п. 2.2.

2.2 Управление входными и выходными сигналами контроллера робота

Загрузить файл модели РТК, который был создан при выполнении индивидуального задания п. 2.1 (рисунок 2.16).

Чтобы исключить повреждение груза в результате удара схватом при неправильном положении груза или в результате падения груза на нижний конвейер при отсутствии палеты, на конвейерах установлены датчики положения с логическими именами **do1** и **do2** (см. рисунок 2.16).

Датчик **do1** выдает значение **1 (истина)**, если груз находится в правильном положении; в этом случае робот может переместить схват в положение захвата груза и закрыть схват для удержания груза. Датчик **do2** выдает значение **1 (истина)**, если палета находится в правильном положении на конвейере; в этом случае робот может переместить схват в положение укладки груза на палету и открыть схват. В остальных случаях выходной сигнал равен **0 (ложь)**.

Для создания программируемых сигналов датчиков откроем вкладку **Offline**, в меню **Offline** на дереве проекта выделим узел контроллера (рисунок 2.17).



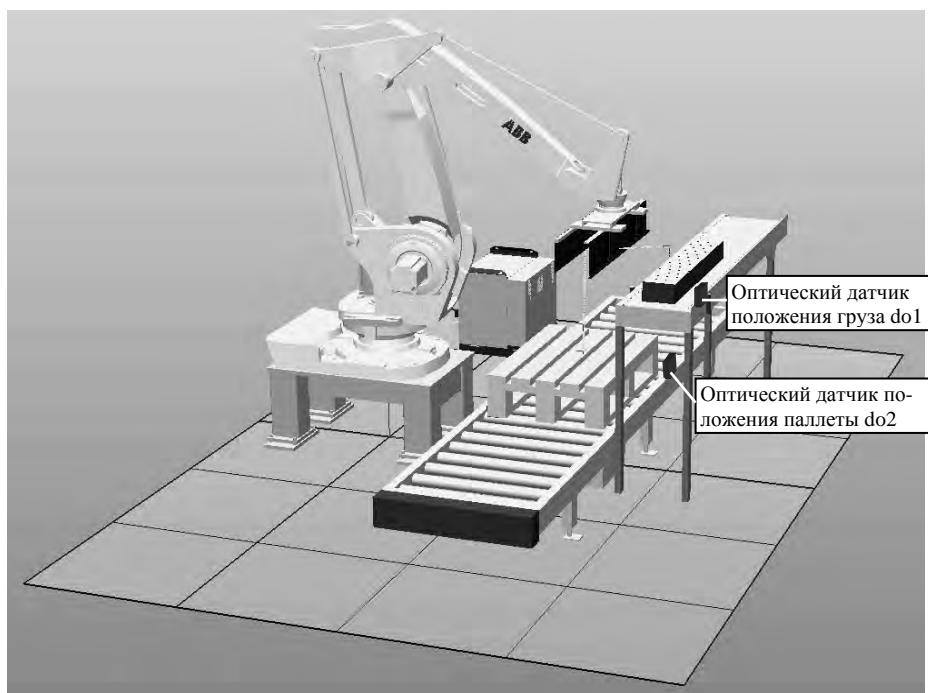


Рисунок 2.16 – Модель РТК и схема расположения датчиков контроля положения груза и паллеты



Рисунок 2.17 – Узел контроллера

Нажмем кнопку **Configuration Configuration Editor**. Выберем пункт **I/O** (input/output, ввод-вывод), в колонке **Type name** – **Signal**. Затем щелкаем правой кнопкой мыши в правой части окна, в контекстном меню выбираем **Add Signal** (добавить сигнал).

Появится окно редактирования сигналов **Edit Signal**. В поле **Name** (имя) вводим имя сигнала **doOpenGripper** (открыть хват), в поле **Type off Signal** (тип сигнала) выбираем **Digital Output** (цифровой выходной), в поле **Access Level** (Уровень Доступа) – **All** (все) и нажимаем **OK**. Аналогично создаем сигнал **doCloseGripper** (закрыть хват). Перезапускаем контроллер командой **Restart** → **Warmstart** («горячий» пуск).

Связываем сигнал с событием (открытие или закрытие схвата). Открываем вкладку **Simulation**, нажимаем кнопку (рисунок 2.18) **Event Manager** (Менеджер Событий).

В появившемся окне нажимаем кнопку **Add**. Появится окно **Choose Trig Type and Activation** (Выберите Тип события и Активацию). Ничего в нем не изменяем, для продолжения нажимаем кнопку **Next**.

В левой части окна **I/O Signal Trigger** (сигнал переключения порта ввода-вывода) в столбце **Signal Name** выбираем сигнал **doOpenGripper**, проверяем, что в правой части окна радиокнопка включена напротив пункта **Signal is true**

(‘1’) (сигнал «истина» =1). Для продолжения нажимаем **Next**.

В окне выбора типа события **Choose Action Type** из выпадающего списка **Set Action Type** (Установить тип события) выбираем пункт **Move Mechanism to Pose** (перемещение механизма в позицию). Для продолжения нажимаем кнопку **Next**.

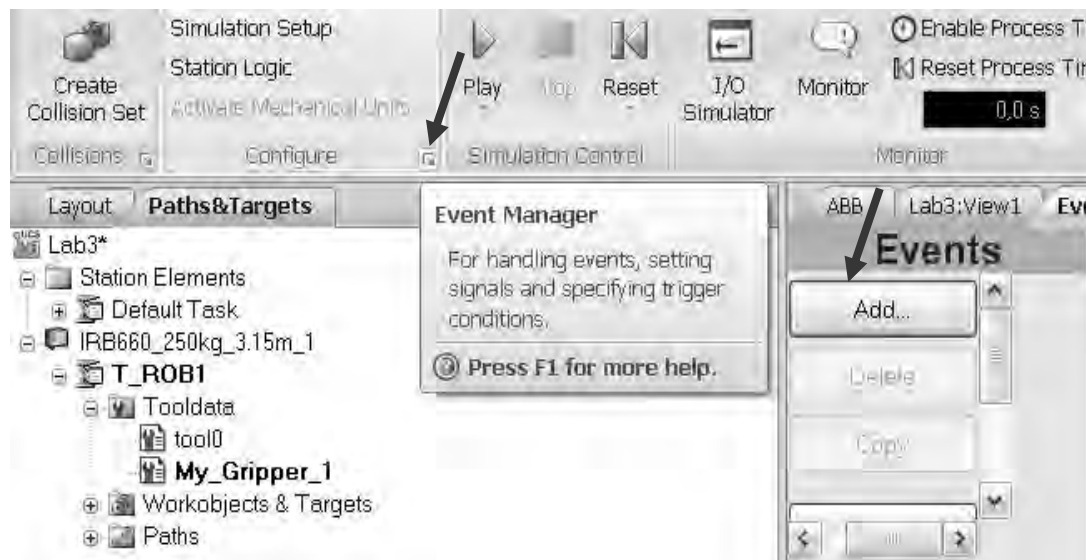


Рисунок 2.18 – Окно кнопка вызова менеджера событий

В окне **Move Mechanism to Pose** в выпадающем списке **Mechanism** выбираем название схвата **My_Gripper**, а в выпадающем списке **Pose** – позицию открытого схвата **HomePose** и нажимаем кнопку **Finish**.

Аналогичным образом выполняем настройку сигнала датчика для открытия схвата **doCloseGripper** с единственным отличием: в последнем окне выбора типа события **Choose Action Type** нажимаем не **Home Pose**, а **Pose Closed**.

Находим вкладку **Simulation**, нажимаем кнопку **I/O Simulator**. В появившемся окне моделирования сигналов портов ввода-вывода (рисунок 2.19) проверим реакцию механизма схвата на изменение сигнала. Чтобы закрыть схват, изменяем значение **doCloseGripper** с **0** на **1**. Для закрытия схвата **doOpenGripper** изменяем значение с **0** на **1**.

Визуализация перемещения груза. Задаем два управляющих сигнала: **Attach** (для прикрепления груза от схвата) и **Detach** (для отделения груза от схвата).

Сигналы создадим в меню **Configuration Editor** аналогично тому, как были созданы сигналы **doCloseGripper** и **doOpenGripper**. Из меню **Configuration Editor** выбираем пункт **I/O** (ввод-вывод), в левой части окна – пункт **Signal**. Заполняем поля окна **Edit Signal(s)** следующим образом: в поле **Name** (имя) вводим имя сигнала **Attach** (присоединить), в поле **Type off Signal** (тип сигнала) выбираем **Digital Output** (цифровой выходной), в поле **Access Level** (Уровень Доступа) – **All** (все) и нажимаем **OK**. Аналогично создаем сигнал **Detach** (отсоединить).

Необходимо создать объект, который будет находиться на схвате и контактировать с грузом, например, полоску из резины для увеличения коэффициента трения между схватом и грузом.

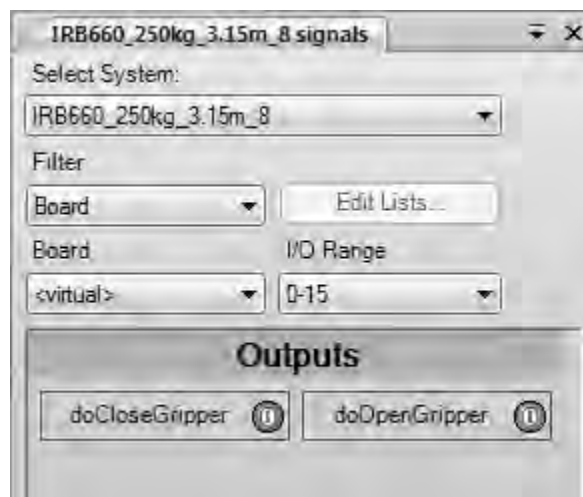


Рисунок 2.19 – Окно моделирования сигналов для портов ввода-вывода контроллера робота

На вкладке **Modeling** (Моделирование) в меню **Surface** (Поверхность) выбираем объект **Surface Rectangle** (Поверхность Прямоугольная).

Габаритные размеры и ориентацию поверхности вводим, как показано на рисунке 2.20. Для завершения создания объекта нажимаем кнопки **Create, Close**.

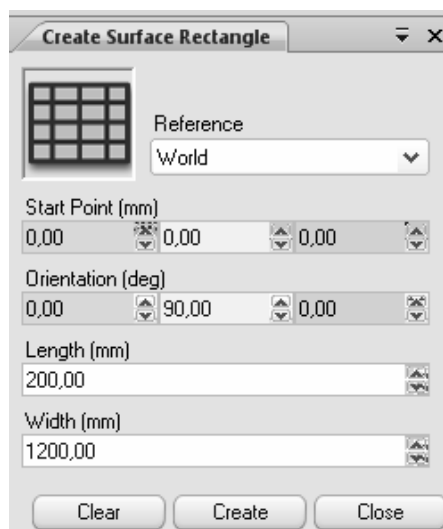


Рисунок 2.20 – Окно создания прямоугольной поверхности заданных размеров и положения

На дереве проекта переименуем созданный объект в **Attacher** и переместим его на захватное устройство. Установим робота в исходное положение, выполнив из контекстного меню команду **Jump Home**.

Свяжем положение объекта **Attacher** и схвата. Выбираем узел **Attacher** в меню **Layout**, в контекстном меню – **AttachTo** → **IRB_660_250_315**.

Установим положение объекта **Attacher**. Выбираем узел **Attacher** в меню **Layout**, в контекстном меню – **Set Position**. Вводим данные, как на рисунке 2.21. Для завершения создания объекта нажимаем кнопки **Apply, Close**.

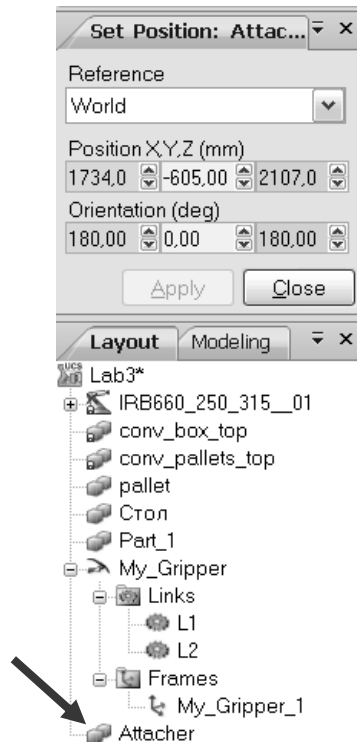


Рисунок 2.21 – Окно определения положения поверхности Attacher

Связываем сигнал с событием. Открываем вкладку **Simulation**, нажимаем кнопку **Event Manager**. В появившемся окне нажимаем кнопку **Add**. Появится окно **Choose Trig Type and Activation**. Ничего в нем не меняем, нажимаем **Next**. В левой части окна **I/O Signal Trigger** выбираем сигнал **Attach**, проверяем, что в правой части окна активна радиокнопка напротив пункта **Signal is true ('1')**. Для продолжения нажимаем **Next**. В окне **Choose Action Type** выбираем пункт **Attach Object**. Для продолжения нажимаем **Next**. Выбираем объект **Груз**, который надо присоединить, и объект, к которому происходит присоединение **Attacher**. Аналогично создаем сигнал **Detach** для отсоединения груза.

Выбираем вкладку **Simulation**, нажимаем кнопку **I/O Simulator**. Кроме сигналов **doCloseGripper** и **doOpenGripper**, на закладке появляются сигналы **Attach** и **Detach**.

При нажатии на кнопку **Attach** груз прикрепляется к захватному устройству. Поэтому при перемещении манипулятора функцией **Jog Joint** груз перемещается вместе с ним.

При нажатии на кнопку **Detach** груз открепляется от захватного устройства. Поэтому при перемещении манипулятора функцией **Jog Joint** груз остается в прежнем положении.

Откроем текст управляющей программы в **Rapid Editor**. Добавим в программу строчки, выделенные полужирным курсивом в тексте программы приведенной ниже.

Текст программы:

```
PROC Path_10()
```

```
MoveJ Home_position,v1000,z100,My_Gripper_1\WObj:=wobj0;
```

```

MoveL take_box_offset,v1000,z100,My_Gripper_1\WObj:=Wobj_box;
MoveL take_box,v1000,fine,My_Gripper_1\WObj:=Wobj_box;
PulseDO doCloseGripper;
PulseDo Attach;
WaitTime 0.5;
MoveL take_box_offset,v1000,z100,My_Gripper_1\WObj:=Wobj_box;
MoveL Home_position,v1000,z100,My_Gripper_1\WObj:=wobj0;
MoveL put_box_offset,v1000,z100,My_Gripper_1\WObj:=Wobj_pallet;
MoveL put_box,v1000,fine,My_Gripper_1\WObj:=Wobj_pallet;
PulseDO doOpenGripper;
PulseDo Detach;
WaitTime 0.5;
MoveL put_box_offset,v1000,z100,My_Gripper_1\WObj:=Wobj_pallet;
MoveJ Home_position,v1000,z100,My_Gripper_1\WObj:=wobj0;
ENDPROC
PROC main()      */главная процедура/*
SetDO doOpenGripper, 0;
SetDO doCloseGripper, 0;
SetDo Attach, 0;      */обнуление значений каналов ввода-вывода/*
SetDO Detach, 0;
PulseDO doOpenGripper;
Path_10; */вызов процедуры задания движения по заданному пути/*
ENDPROC

```

Добавленные команды имеют следующее назначение:

SetDO – устанавливает значение сигнала (1 – истина, 0 – ложь);

PulseDO – подает управляющий импульс на привод механизма (схвата);

WaitTime – задержка исполнения следующей команды для выделения времени на срабатывание механизма, сек.

Для сохранения отредактированного текста программы нажимаем кнопку **Apply Changes** (сохранить изменения) и закрываем окно редактора.

Для моделирования движения робота по измененной программе в меню **Simulation Control** (Контроль Моделирования) нажимаем кнопку **Play**. В графическом окне системы наблюдаем, как робот перемещается по оптимальному пути. В итоге захватное устройство закрывается, когда нужно взять груз, и открывается, когда нужно отпустить его.

Индивидуальное задание.

Создать сигналы для портов ввода-вывода от датчиков контроля положения грузов и выполнить визуализацию операции загрузки конвейера для РТК, который был разработан при выполнении индивидуального задания по п. 2.1 практической работы № 2.



2.3 Цифровое прототипирование технологического оборудования РТК и ГПС

Цифровой прототип – это цифровой макет изделия, используемый для анализа формы и функциональности изделий до их изготовления. Цифровой прототип становится все более совершенным по мере того, как интегрируются все концептуальные, механические и электрические проектные данные.

Полный цифровой прототип является виртуальным опытным образцом готового изделия и служит для его оптимизации и проверки. Это снижает потребность в изготовлении дорогостоящих физических опытных образцов.

2.3.1 Цифровое прототипирование металлорежущих станков. Все наиболее распространенные в практике машиностроительного проектирования САД-системы (AutoCAD, Inventor, SolidWorks, и др.) обеспечивают создание трехмерных цифровых прототипов изделий и располагают функциональными возможностями их всестороннего анализа.

Загружаем систему AutoCAD из главного меню Windows и создаем новый чертеж на основе шаблона **acadiso.dwt**.

На первом этапе моделирования на основе размеров отдельных узлов станка (рисунки 2.22–2.24) создаем 3D-модель станка путем выдавливания контура.

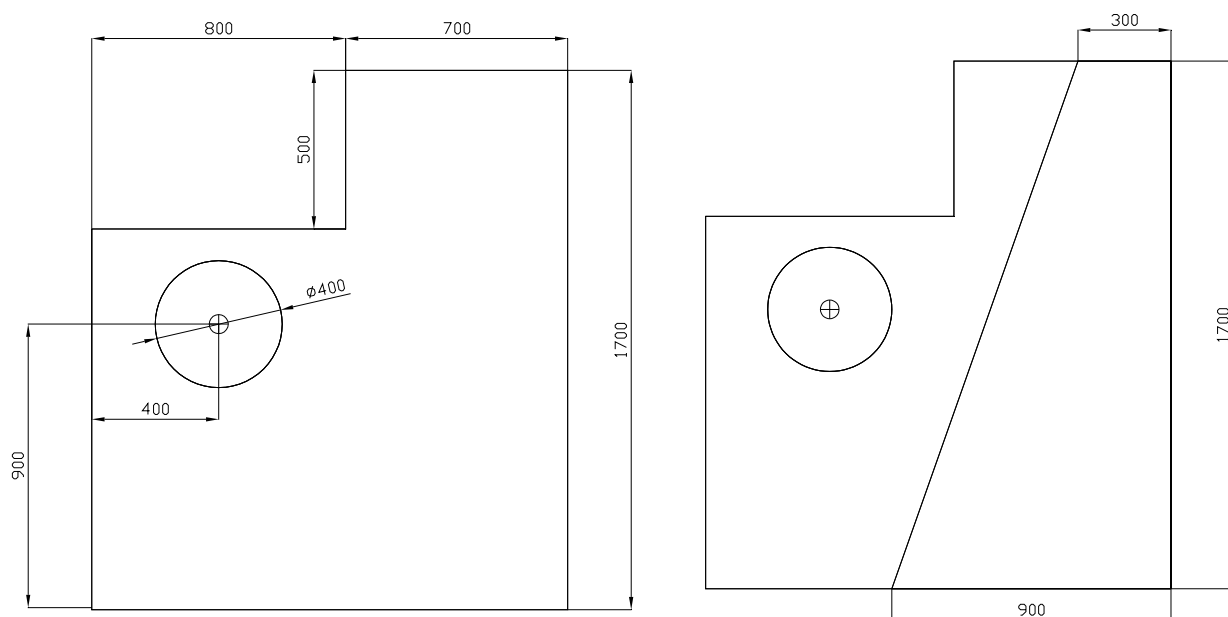


Рисунок 2.22 – Набор контуров для создания 3D-модели станка

Выполняя раскрашивание и вращение модели в пространстве, оцениваем результаты моделирования (рисунок 2.25).

При моделировании механизмов в среде RobotStudio необходимо, чтобы подвижные и неподвижные части модели хранились в отдельных файлах.

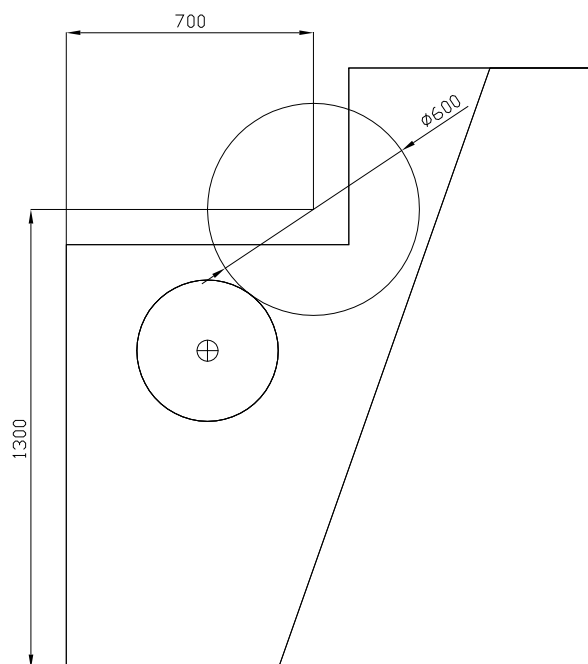


Рисунок 2.23 – Положение и размеры модели инструментального магазина станка

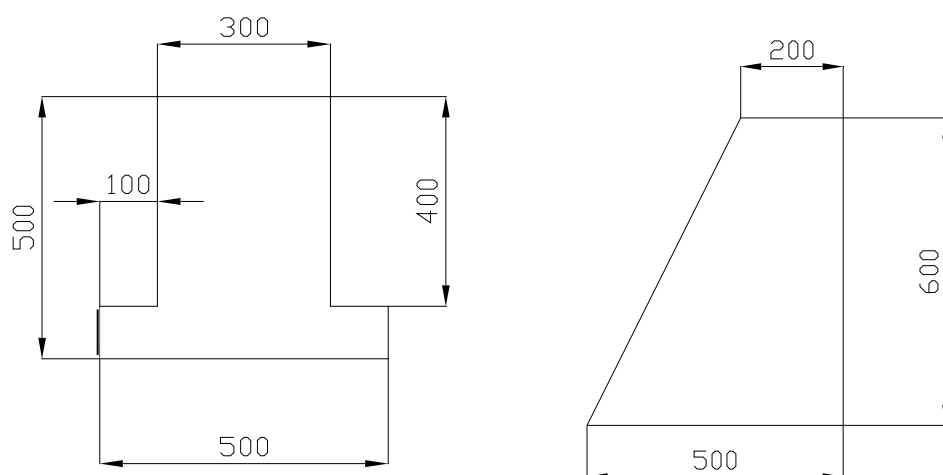


Рисунок 2.24 – Размеры контура привода револьверной головки, суппорта и задней бабки станка

2.3.2 Цифровое прототипирование обрабатываемой заготовки и кассеты. В рабочей папке создаем файл **Rolling.dwg**. Для создания цифрового прототипа заготовки из проката с центром в точке сначала вычерчиваем цилиндр радиусом 30 мм и высотой 200 мм (рисунок 2.26). Формируем в модели центровые отверстия.

Сохраняем в своей рабочей папке файл Заготовка.dwg, а затем командой **Файл** → **Экспорт** создаем файл экспорта под именем **Rolling.sat**.

Таким же образом построим цифровой прототип кассеты для размещения заготовки на шаговом конвейере. В рабочей папке создаем файл **Cartridge.dwg**. Чертеж кассеты выдает преподаватель. Общий вид кассеты приведен на рисунке 2.27.

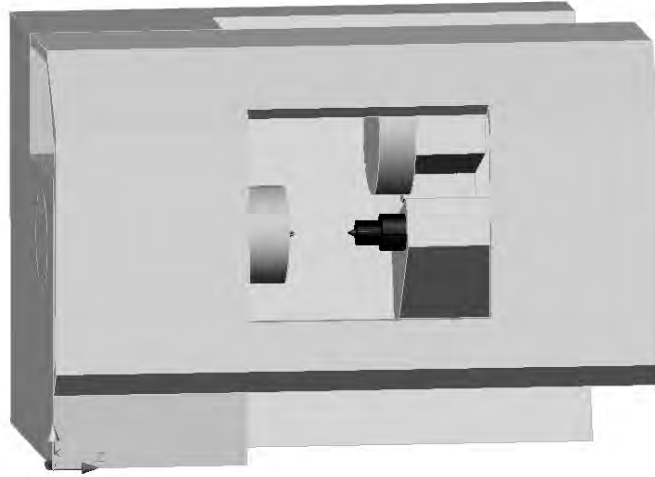


Рисунок 2.25 – Результаты моделирования кабины станка



Рисунок 2.26 – Модель заготовки с центровыми отверстиями

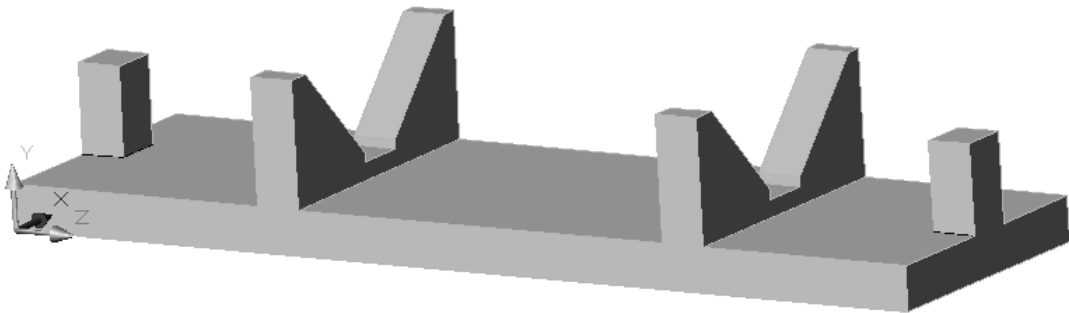


Рисунок 2.27 – Кассета для размещения заготовки на шаговом конвейере

Сохраняем в своей рабочей папке файл **Cartridge.dwg**, а затем командой **Файл → Экспорт** формируем файл экспорта под именем **Cartridge.sat**.

2.3.3 Цифровое прототипирование схвата. В рабочей папке создаем файл **Gripper.dwg**. Строим контур поперечного сечения левого призматического рычага схвата по размерам, показанным на рисунке 2.28.

Выполняя раскрашивание и вращение модели в пространстве, оцениваем результаты моделирования (рисунок 2.29). Сохраняем в своей рабочей папке файл **Gripper.dwg**.

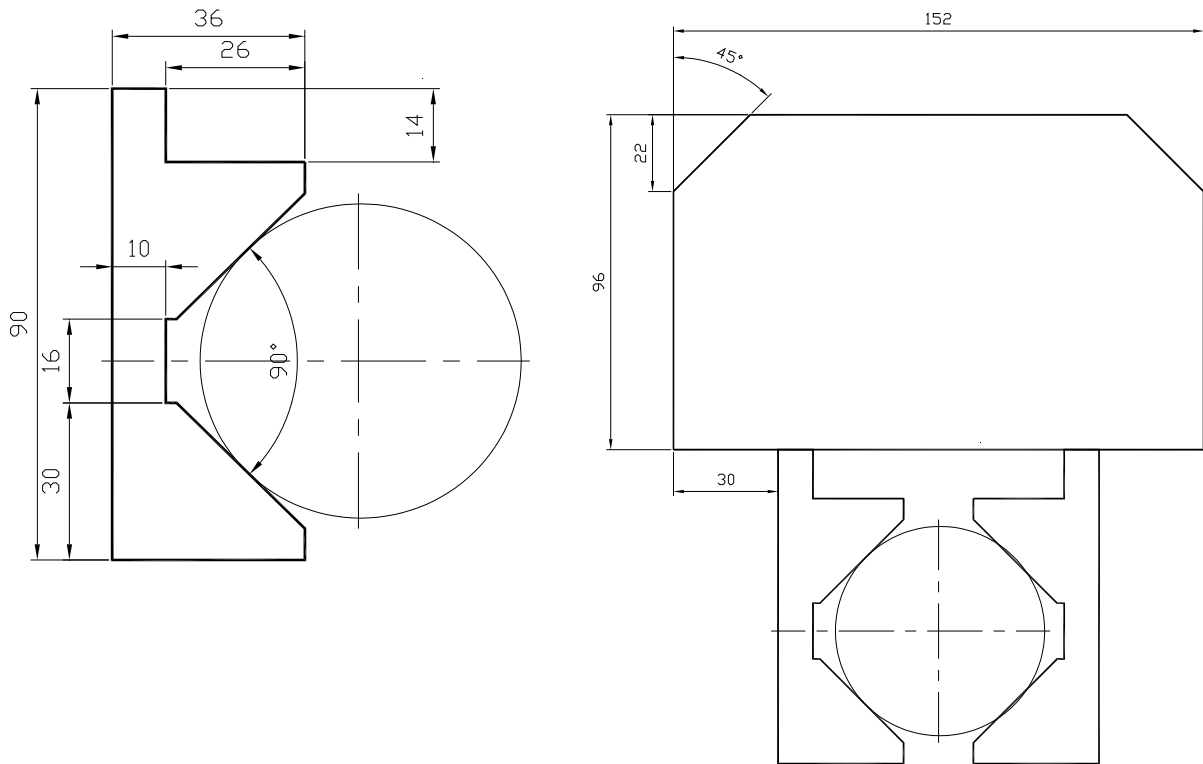


Рисунок 2.28 – Контур поперечного сечения левого призматического рычага схвата и корпуса схвата

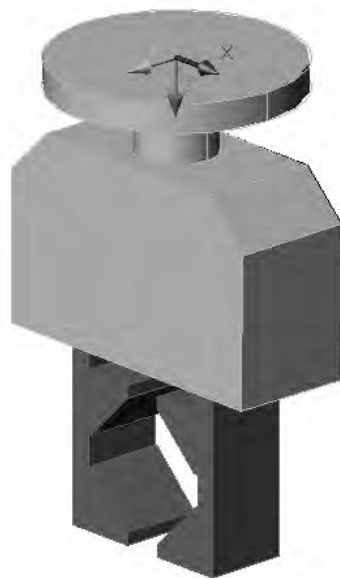


Рисунок 2.29 – Результат моделирования схвата

Для последующего моделирования работы механизма схвата сохраняем подвижные и неподвижные элементы схвата в разных файлах экспорта.

2.3.4 Создание механизмов «Дверь» и «Пиноль». Создаем пустую станцию командой **Empty Station**. Командой **Import Geometry** (меню Home) загружаем файл **Lathe_Bed.sat**, а затем файлы **Door.sat**, **Pinole.sat** из своей рабочей папки.

Командой **Set Position** последовательно поворачиваем станину, дверь и пиноль на 90 град вокруг оси X и 180 град вокруг оси Z .

Из меню **Modeling** выполняем команду **Create Mechanism** (создать механизм). В окне **Create Mechanism** в поле **Mechanism Model Name** (имя модели механизма) вводим **Lathe** (токарный станок). Выбираем из выпадающего списка **Mechanism Type** строку **Device** (Устройство).

На дереве **Lathe** в окне **Create Mechanism** выбираем узел **Links** (связи) и из контекстного меню узла выполняем команду **Add Link** (добавить связь). В окне **Create Link** (создать связь) в поле **Selected Part** (выбрать часть) выбираем **Lathe**, устанавливаем флаг **Set as BaseLink** (база, неподвижная часть) и жмем зеленую стрелку. Нажимаем кнопку **Apply** (применить). В поле **Selected Part** выбираем **Door**, кликаем мышью зеленую стрелку. Нажимаем кнопку **Apply**. В поле **Selected Part** выбираем **Pinole**, жмем зеленую стрелку и последовательно кнопки **Apply**, **Cancel**. Таким образом, определен состав подвижных и неподвижных деталей механизма.

Для задания перемещения двери на дереве **Lathe** выбираем узел **Joints** (перемещения). Из контекстного меню узла выполняем команду **ADD Joint** (добавить перемещение). В окне (рисунок 2.30) **Create Joint** (создать перемещение) выбираем тип перемещения **Prismatic** (линейное).

Рисунок 2.30 – Окно задания величины перемещений

В поле **Second Position** (вторая позиция) вводим координату Y 1 мм (определяем направление перемещения вдоль оси Y). В поле **Jog Axis** (перемещение по оси) перемещаем движок и наблюдаем перемещение двери в окне кабины станка. В поле **Min Limit** (минимальный предел перемещения) записываем 0, в поле **Max Limit** (максимальный предел перемещения) – 1000 мм. Нажимаем кнопку **Apply** (применить).

Для задания перемещения пиноли в поле **Child Link** (дочерняя связь) выбираем связь $L3$. В окне **Create Joint** выбираем тип перемещения **Prismatic**. В поле **Second Position** вводим координату Y 1 мм (определяем направление перемещения вдоль оси Y). В поле **Min Limit** (минимальный предел перемещения) записываем 0, в поле **Max Limit** (максимальный предел перемещения) – 16 мм. Последовательно нажимаем кнопки **Apply**, **Cancel**. Внизу окна **Create Mechanism** нажимаем кнопку **Compile Mechanism** (Собрать механизм).

Открываем закладку **Layout**, выбираем узел **Door**. В окне **Create Mechanism** щелкаем кнопку **Add**. В окне (рисунок 2.31) **Create Pose** (создать положение) в поле **Pose Name** (имя положения) вводим имя положения двери (движок стоит в положении 00) **Door Closed** (Дверь закрыта) и нажимаем кнопку **Apply**. В окне **Create Pose** вводим имя положения двери (движок стоит в положении 1000) **DoorOpened** (Дверь открыта) и нажимаем кнопку **Ok**.

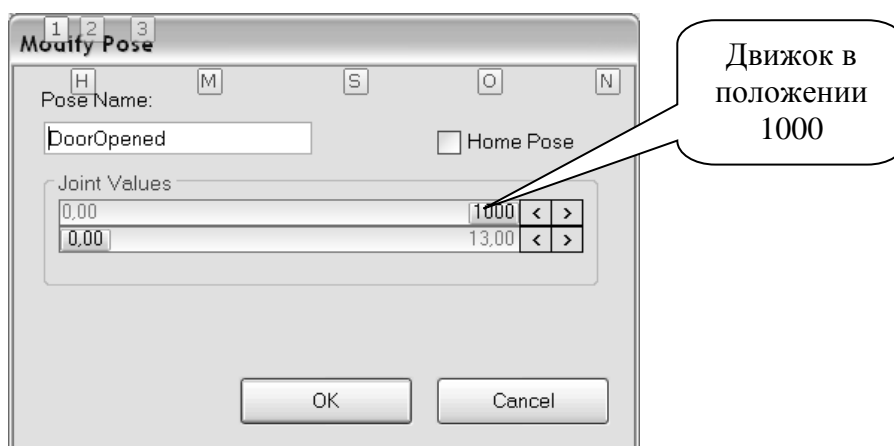


Рисунок 2.31 – Окно задания положения подвижных элементов механизма

Аналогичным образом задаем положения пиноли задней бабки станка **PinoleOn** (пиноль подведена), **PinoleOff** (пиноль отведена), используя движок для задания положения пиноли.

При моделировании РТК может быть установлена длительность цикла выполнения операции. Для того чтобы это значение было корректно определено, задается время перемещения частей моделируемого механизма. Для этого нажимаем кнопку **Set Transition Times** (установить время перемещения) и в соответствующих полях (рисунок 2.32) **Door Closed**, **DoorOpened** устанавливаем время срабатывания механизма на открытие и закрытие дверей, например, 2 с. Аналогичным образом задаем время отвода **PinoleOff**, подвода **PinoleOn** пиноли задней бабки станка – 1 с. Закрываем последовательно окна **Set Transition Times**, **Create Mechanism** командой **Close**. На запрос «Are you sure that you

want to close Mechanism Modeler?» (Вы действительно желаете завершить моделирование механизма) отвечаем «Да».

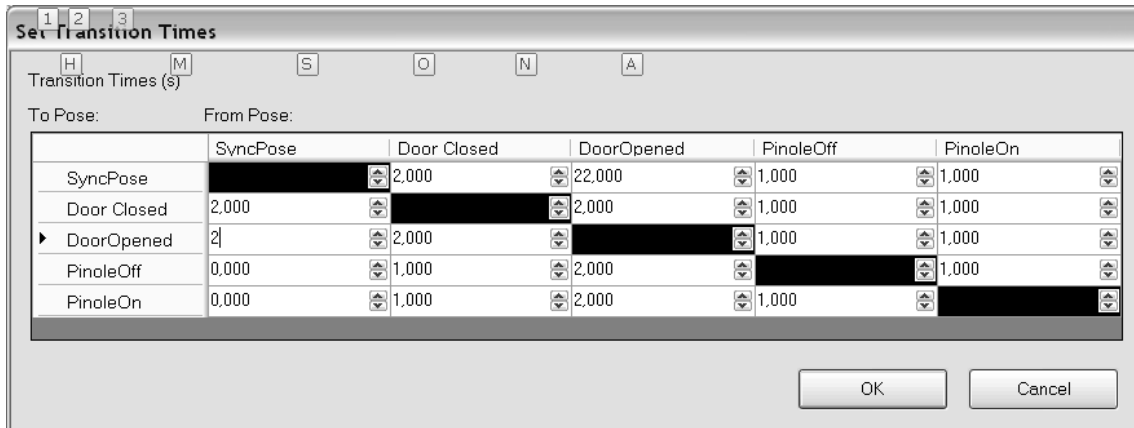


Рисунок 2.32 – Задание длительности перемещения подвижных частей механизма

На дереве проекта (закладка Layout) выбираем **Lathe**, открываем закладку меню **Modify** и открываем меню **Move to Pose** (переместить в положение). Поочередно выбираем пункты **Door Closed**, **DoorOpened**, **PinoleOff**, **PinoleOn**, наблюдаем закрытие, открытие двери и перемещение пиноли.

На дереве проекта выбираем узел **Door**, затем из контекстного меню – команду **Mechanism Joint Jog**, на поле окна перемещаем движок, наблюдаем открытие-закрытие двери.

На дереве проекта выбираем узел **Lathe** и из контекстного меню узла выполняем команду **Save as Library** (сохранить в библиотеке), вводим имя файла **Lathe** и сохраняем файл с расширением ***.rslib** как библиотечный файл Robot Studio в своей рабочей папке. В дальнейшем созданный цифровой прототип токарного станка может быть использован при создании различных РТК и их программировании.

2.3.5 Создание механизма «Схват». Создаем пустую станцию командой **Empty Station**. Командой **Import Geometry** (меню Home) загрузим файл корпуса схвата **Mounting.sat**, а затем файлы рычагов схвата **Lever_R.sat**, **Lever_L.sat** из своей рабочей папки.

Далее в соответствии с методикой, изложенной выше, создаем механизм **Gripper** (схват) с типом **Tool** (инструмент), который состоит из неподвижной части **Mounting** (корпус) и подвижных частей, правого и левого рычагов **Lever_R**, **Lever_L**.

В ходе создания механизма **Gripper** для правого и левого рычагов **Lever_R**, **Lever_L** должны быть сформированы положения **Lever_R_Off** (закрыт), **Lever_R_On** (открыт), **Lever_L_Off** (закрыт), **Lever_L_On** (открыт). Величину перемещения рычагов схвата установите, перемещая движок положения моделируемой подвижной части механизма.

При создании на дереве механизма определим данные инструмента **ToolData**. Масса схвата с заготовкой – 7,5 кг. Время срабатывания подвижных

частей схвата установим равным 1 с.

После завершения создания механизма и проверки его работы сохраним его в файле под именем **Gripper** с расширением ***.rslib** как библиотечный файл RobotStudio в своей рабочей папке.

В результате цифрового прототипирования основного и вспомогательного технологического оборудования созданы цифровые прототипы, которые отражают не только геометрические параметры моделируемых объектов, но и такие их характеристики, как величины перемещений подвижных элементов оборудования, скорости их перемещения, массоинерционные свойства.

2.4 Общие сведения о контроллере IRC-5 и офлайн-программировании РТК на языке RAPID

Управление роботами (рисунок 2.33) производства компании ABB Group (Швеция) обеспечивается блоком модулей управления (Control module), связанным с силовыми блоками управления отдельными роботами (Drive module). Блок модулей управления обеспечивает управление синхронной работой нескольких роботов (до 4-х) с помощью функции **MultiMove**.

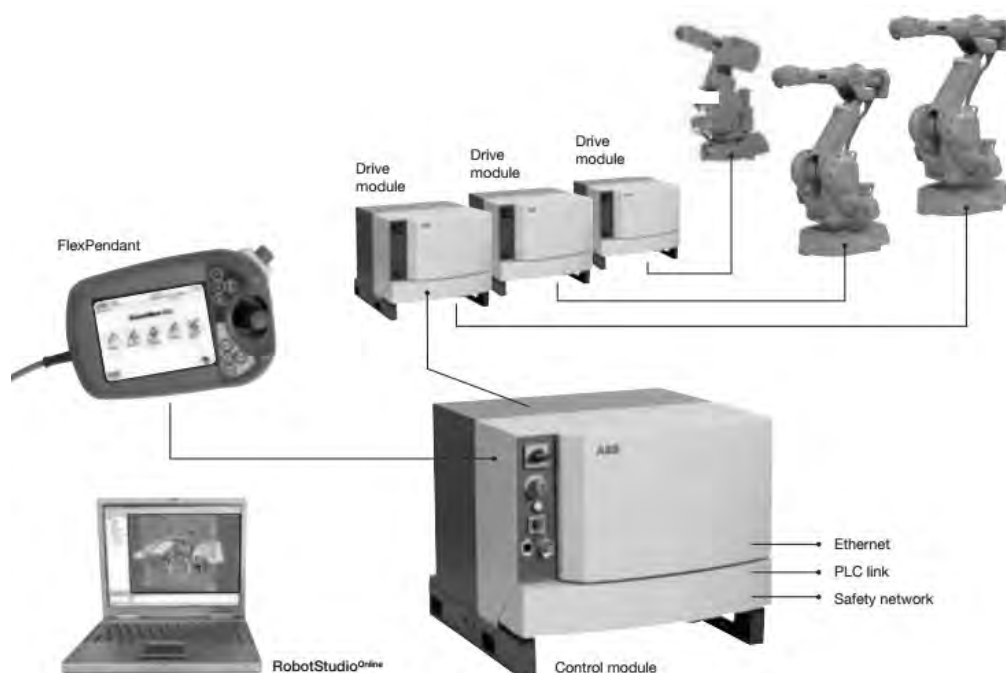


Рисунок 2.33 – Блок модулей управления (Control module) роботами IRC5

При создании блока модулей управления пятого поколения использована новая концепция модульного проектирования (рисунок 2.34).

Модульная конструкция блока управления (Control module) обеспечивает управление не только роботами, но и технологическим процессом. Подобная гибкость дает возможность создавать оптимальные компоновки гибких РТК, а также выполнять модернизацию или производить замену одного из модулей при минимальном вмешательстве в другие модули.

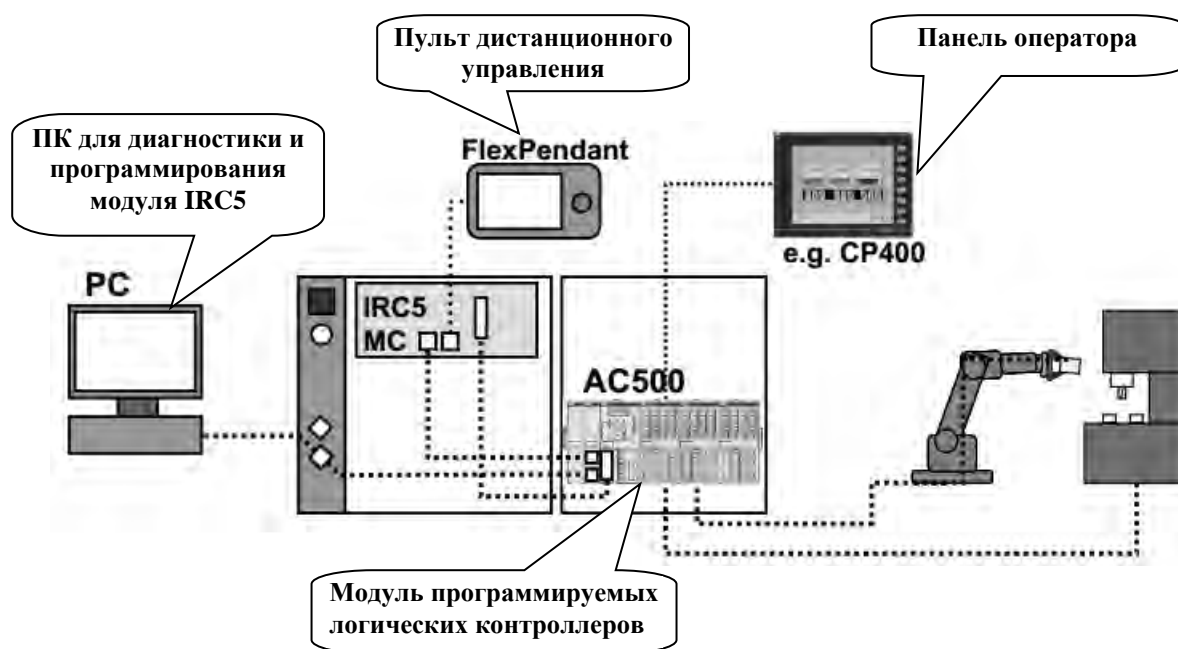


Рисунок 2.34 – Общая концепция использования модуля IRC5 для управления роботом и технологическим оборудованием

Для исключения простоя оборудования при программировании роботов на выполнение других операций к модулю IRC5 (рисунок 2.35) может быть подключен персональный компьютер для офлайн-программирования РТК на языке RAPID.

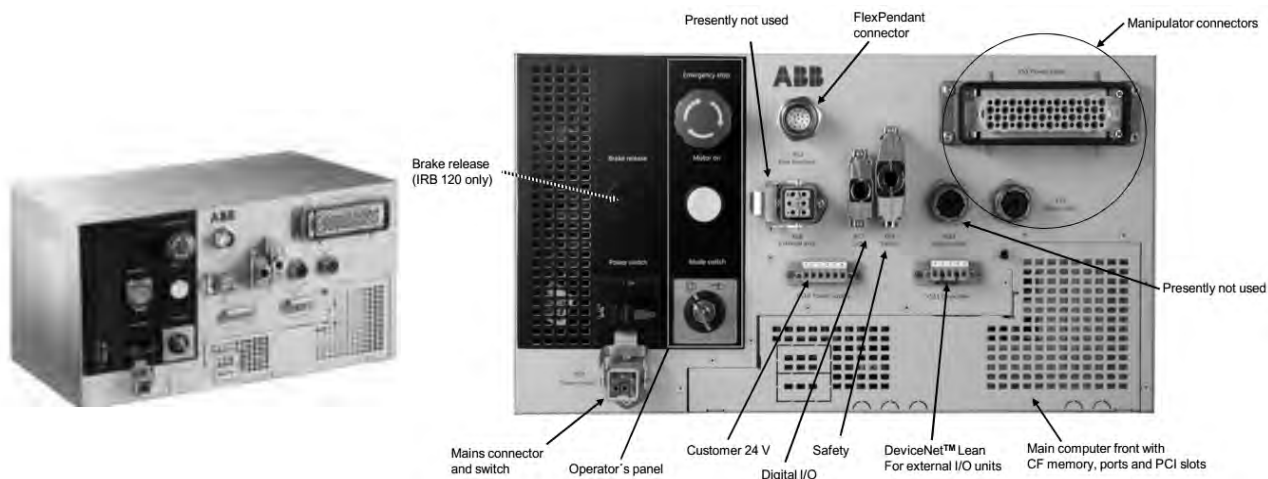


Рисунок 2.35 – Модуль управления IRC5

Для управления технологическим оборудованием и анализа его состояния к блоку управления могут быть подсоединены различные аналоговые и цифровые устройства, которые обеспечивают включение и выключение отдельных механизмов станков, конвейеров и выполняют оценку состояния этих механизмов после исполнения управляющих команд. Для этих целей блок управления стандартной комплектации имеет 16 портов ввода-вывода цифровых сигналов (Digital I/O 16 DI/16 DC), 8 портов релейных сигналов для включения

контакторов и реле (8 DI /8 DO relay) и 4 аналоговых канала ввода-вывода (Analog 4 AI/4 AO) (рисунок 2.36).

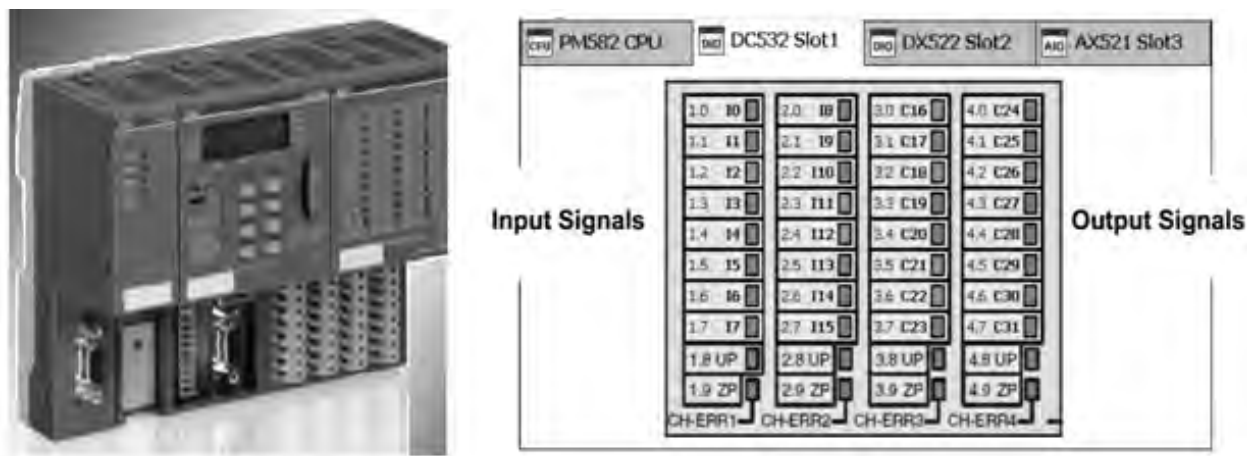


Рисунок 2.36 – Коммуникационный интерфейс портов ввода-вывода модуля программируемого логического контроллера AC500

Программирование модуля управления IRC5 в условиях гибкого производства для исключения простоев оборудования выполняется в режиме офлайн с использованием виртуального контроллера. Для программирования РТК, создаваемых на базе роботов производства ABB Group, используется специальный язык программирования RAPID. С целью сокращения трудоемкости программирования гибких роботизированных комплексов концерном ABB Group разработана система моделирования и программирования РТК RobotStudio.

RAPID-программа содержит данные и инструкции, которые управляют роботом и периферийным оборудованием. RAPID-программа (рисунок 2.37) обычно состоит из четырех частей.

1 Главная программа, которая вызывает для выполнения подпрограммы.

2 Несколько подпрограмм, которые разделяют программу на меньшие части, вызываемые из главных или других подпрограмм.

3 Условно-постоянные данные программы, которые используются для определения положения элементов РТК, числовых значений и систем координат.

4 Системные модули – программы, которые постоянно находятся в оперативной памяти и предназначены для управления исполнительными механизмами робота.

При моделировании производственного процесса в среде RobotStudio формирование программы на языке RAPID происходит автоматически. При необходимости в программу могут быть внесены изменения с использованием редактора системы программирования RAPID Editor (рисунок 2.38).

Как и большинство языков программирования высокого уровня, RAPID содержит функции определения переменных, массивов и констант (Var, Dim, Const), математические функции, функции организации циклов и условных переходов (While, Do, For, If, GoTo), синтаксис которых подобен синтаксису языка Basic.

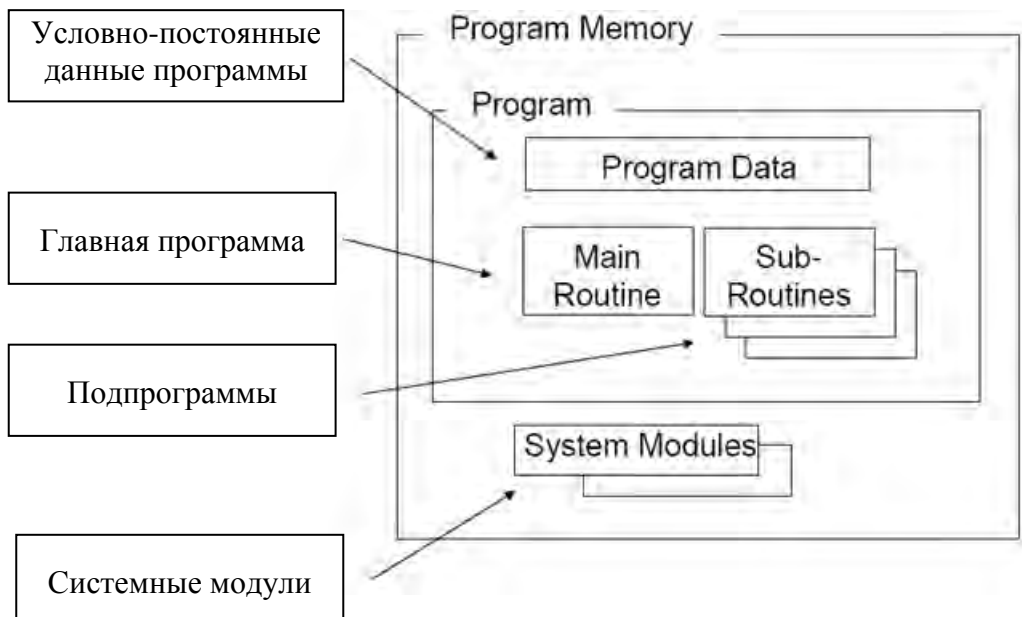


Рисунок 2.37 – Структура RAPID-программы

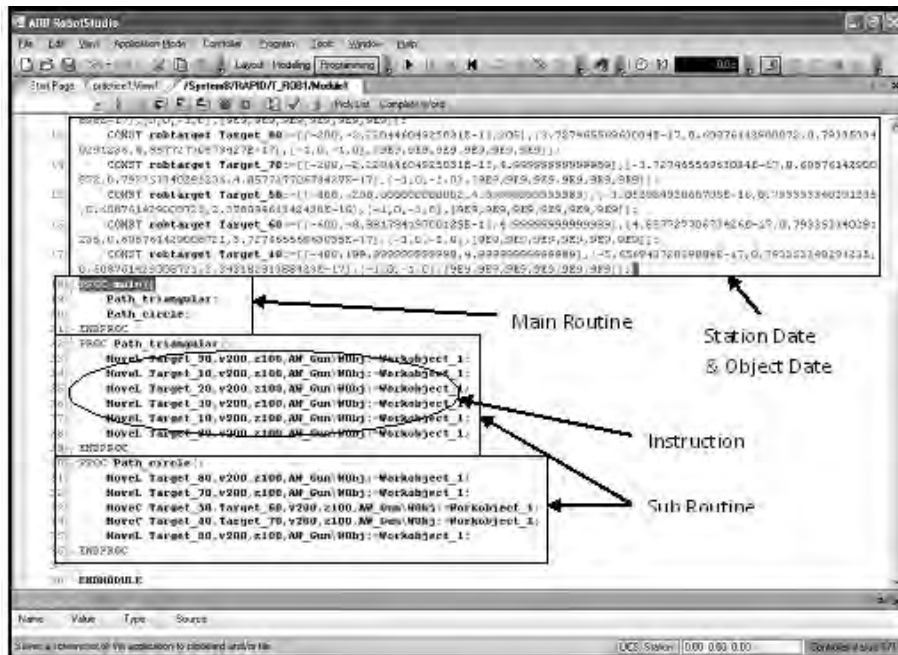


Рисунок 2.38 – Программный интерфейс RAPID Editor

Специфическими особенностями языка RAPID является наличие в нем специальных функций – инструкций для управления перемещениями руки робота, периферийного оборудования и обработки сигналов цифровых и аналоговых датчиков.

Для задания перемещений руки робота используются инструкции позиционирования. Их основные типы приведены далее.

Инструкция **MoveL** используется, чтобы переместить точку центра инструмента (**TCP – Tool Center Point**) из исходной точки в заданную по прямой, соединяющей эти точки. Синтаксис инструкции имеет вид:

MoveL [\Conc] ToPoint Speed [\V] | [\T] Zone [\Z] Tool [\WObj]

Инструкция **MoveJ** используется, чтобы переместить руку робота кратчайшим путем (за наименьшее время) из одного положения в другое, при этом все звенья руки робота завершают движение одновременно. Синтаксис инструкции имеет вид:

MoveJ [\Conc] ToPoint Speed [\V] | [\T] Zone [\Z] Tool [\WObj]

Инструкция **MoveC** используется, чтобы переместить точку центра инструмента (TCP), в заданную точку по дуге окружности. Синтаксис инструкции имеет вид:

MoveC [\Conc] CirPoint ToPoint Speed [\V] | [\T] Zone [\Z] Tool [\WObj]

Параметры инструкций перемещения имеют следующее назначение.

[\Conc] – TCP инструмента перемещается линейно согласно заданной инструкции с одновременным выполнением логических инструкций.

ToPoint – конечная точка перемещения TCP.

CirPoint – точка, лежащая на дуге окружности между точками ее начала и конца.

Speed [\V] – скорость перемещения TCP в мм/с, например, v1000.

[\T] – заданное время перемещения инструмента, например, v1000\T: = 5.

Полное движение занимает 5 с.

Zone [\Z] – задается для обхода промежуточных точек траектории, чтобы в этих точках не происходил останов перемещения робота в связи с резким изменением направления движения.

Tool [\WObj] – имя рабочего объекта инструмента.

Квадратные скобки указывают на необязательное задание параметра.

Управление оборудованием и обработка сигналов датчиков выполняется с использованием следующих инструкций.

PulseDO [\PLength] – определяет продолжительность выходного импульсного сигнала, например, PulseDO \PLength: = 1.0. Генерирует сигнал продолжительностью 1 с.

Reset Signal – используется для очистки порта и установки выходного сигнала в 0, например Reset do15.

Set Signal – используется для инициализации выходного порта и установки его сигнала в 1, например, Set do15.

SetAO Signal Value – устанавливает величину выходного аналогового сигнала, например, SetAO ao2, 5.5.

SetDO [\SDelay] Signal Value – изменяет значение выходного цифрового сигнала, например, SetDO do15, 1.

WaitDI Signal Value [\MaxTime] [\TimeFlag] – используется для задержки, пока входной цифровой сигнал не установится в заданное значение, например, WaitDO di4, 1.



WaitDO Signal Value [MaxTime] [TimeFlag] – используется для задержки, пока выходной цифровой сигнал не установится в заданное значение, например, WaitDO do4, 1.

WaitTime [InPos] Time – задает задержку времени до исполнения следующей инструкции в секундах. Например, ожидание завершения операции, выполняемой на периферийном оборудовании.

Список литературы

1 Захватные устройства промышленных роботов и манипуляторов / А. А. Москвичев, А. Р. Кварталов, Б. В. Устинов. – Москва : ФОРУМ ; ИНФРА-М, 2015. – 176 с.

2 **Иванов, А. А.** Основы робототехники : учебное пособие / А. А. Иванов. – Москва : Форум, 2014. – 224 с.

3 **Выжигин, А. Ю.** Гибкие производственные системы : учебное пособие для вузов / А. Ю. Выжигин. – Москва : Машиностроение, 2009. – 288 с.

4 **Схиртладзе, А. Г.** Автоматизация производственных процессов в машиностроении : учебник для вузов / А. Г. Схиртладзе, В. Н. Воронов, В. П. Борискин. – Старый Оскол : ТНТ, 2011. – 612 с.

5 **Фельдштейн, Е. Э.** Автоматизация производственных процессов в машиностроении : учебное пособие / Е. Э. Фельдштейн, М. А. Корниевич. – Москва : ИНФРА-М ; Новое знание, 2016. – 264 с.

6 Роботизированные производственные комплексы / Под ред. Ю. Г. Козырева, А. А. Кудинова. – Москва : Машиностроение, 1987. – 272 с.

7 **Довбня, Н. М.** Роботизированные технологические комплексы в ГПС / Н. М. Довбня, А. Н. Кондратьев, Е. И. Юревич. – Ленинград : Машиностроение, 1990. – 303 с.

