

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Технология машиностроения»

ОСНОВЫ МЕХАТРОНИКИ И РОБОТОТЕХНИКИ

*Методические рекомендации к практическим занятиям
для студентов направления подготовки
15.03.06 «Мехатроника и робототехника»
дневной формы обучения*



Могилев 2023

УДК 621.3+621.865.8
ББК 32.816
О 75

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Технология машиностроения» «28» сентября 2023 г.,
протокол № 2

Составитель О. Н. Кляус
Рецензент Д. М. Свирепа

Даны рекомендации для проведения практических занятий, целью которых является изучение принципов построения и основ расчета робототехнических мехатронных систем на базе учебно-исследовательского комплекса и промышленного робота Кавасаки. Изложена последовательность расчета, управления и программирования основных технических параметров робота. Сформулированы вопросы и задания.

Учебное издание

ОСНОВЫ МЕХАТРОНИКИ И РОБОТОТЕХНИКИ

| | |
|-------------------------|-----------------|
| Ответственный за выпуск | В. М. Шеменков |
| Корректор | И. В. Голубцова |
| Компьютерная верстка | М. М. Дударева |

Подписано в печать. 29.12.2023. Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л 2,79. Уч.-изд. л. 3,0 . Тираж 31 экз. Заказ № 1415.

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2023

Содержание

| | |
|--|----|
| 1 Практическое занятие № 1. Прямые и обратные задачи управления роботами. Практическое занятие № 2. Изучение структуры и компонентов учебного роботизированного комплекса..... | 4 |
| 2 Практическое занятие № 3. Позиционные и кинематические задачи управления роботами. Практическое занятие № 4. Изучение методов ручного управления роботом..... | 15 |
| 3 Практическое занятие № 5. Управление схватом манипуляционного робота по вектору скорости. Практическое занятие № 6. Блочное программирование робота..... | 19 |
| 4 Практическое занятие № 7. Изучение языка программирования AS робота Kawasaki. Разработка программ..... | 26 |
| 5 Практическое занятие № 8. Изучение функций ввода команд программ с помощью редактора KRterm. Отладка и выполнение программ роботом..... | 34 |
| 6 Практическое занятие № 9. Изучение среды программирования K-ROSET робота Kawasaki..... | 43 |
| Список литературы..... | 48 |

1 Практическое занятие № 1. Прямые и обратные задачи управления роботами. Практическое занятие № 2. Изучение структуры и компонентов учебного роботизированного комплекса

Учебно-исследовательский комплекс на базе промышленного робота Kawasaki.

В состав комплекса входят: манипулятор Kawasaki FS003N; контроллер Kawasaki FD70; пульт оператора; вакуумный схват (установлен на манипулятор); монтажная панель; программа компьютерной симуляции Kawasaki PC ROSET (рисунок 1.1).



Рисунок 1.1 – Учебно-исследовательский комплекс

Дополнительное оборудование – система технического зрения: видеокамера 2D; программное обеспечение обработки изображения RFA-Vision v4.1; ноутбук.

Описание контроллера.

1 Панель контроллера.

Внешняя передняя панель контроллера или системы управления (СУ) робота осуществляет следующие функции (рисунок 1.2, таблица 1.1).

2 Включение силового питания контроллера.

Нужно переместить вправо переключатель силового питания (номер позиции 1, см. таблицу 1.1), расположенный в нижней правой части панели контроллера.

3 Включение и выключение силового питания привода робота.

Для включения силового питания двигателя робота нужно нажать клавиши A + Motor ON на пульте ручного управления. Затем нажать клавиши A + + RUN для разрешения движения. При этом окна MOTOR и RUN будут

включены на экране пульта управления. Выключение силового питания привода осуществляется в обратном порядке: A + Motor ON, затем A + + RUN или HOLD – останавливает движение.

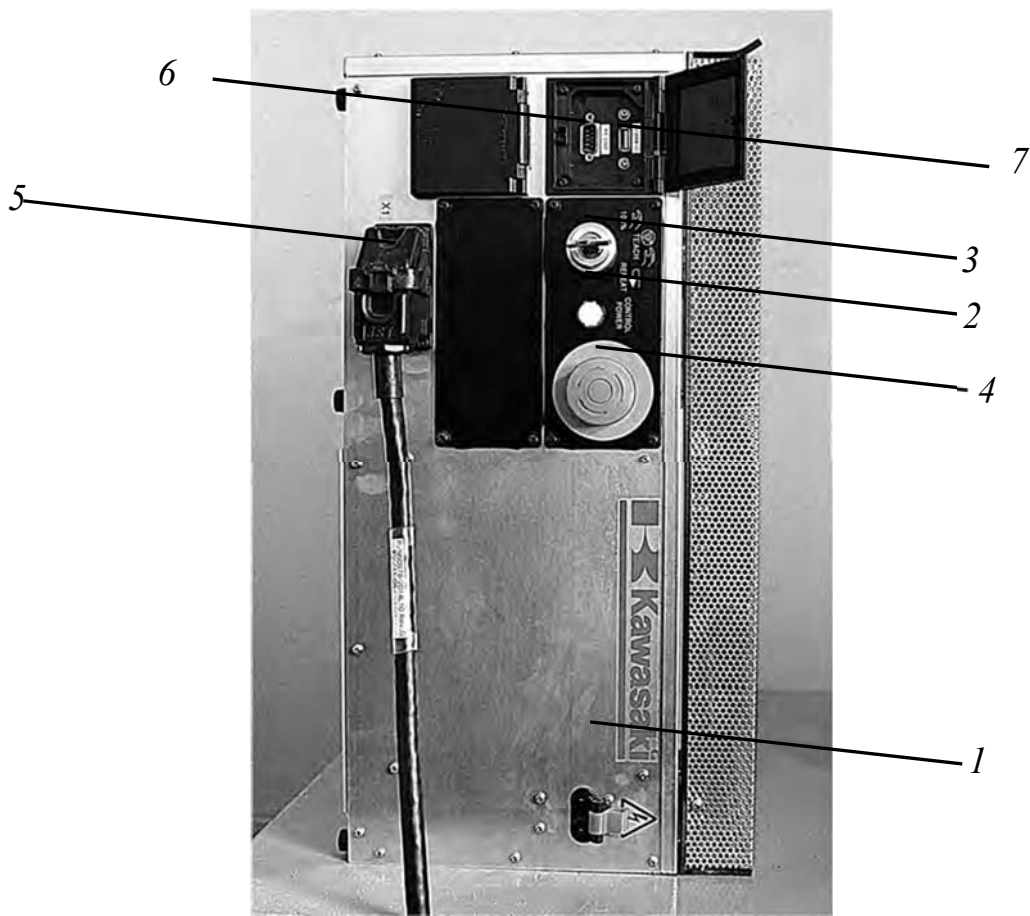


Рисунок 1.2 – Панель контроллера

Таблица 1.1 – Переключатели и лампочки на передней панели контроллера. Внешние входы/выходы на передней панели контроллера

| Номер позиции (см. рисунок 1.2) | Переключатель и лампочки | Назначение |
|---------------------------------|--------------------------------------|---|
| 1 | 2 | 3 |
| 1 | Переключатель | Включает и выключает общее силовое питание |
| 2 | Индикатор источника силового питания | Загорается при включении общего силового питания |
| 3 | Teach/Repeat переключатель | Переключает режимы «обучение» или «автоматический режим». Обеспечивает выбор режимов управления (режим обучения, когда переключатель находится в горизонтальном положении TEACH; режим выполнения программ, когда переключатель повернут по часовой стрелке и находится в положении REPEAT) |

Окончание таблицы 1.1

| 1 | 2 | 3 |
|---|--------------------------------|--|
| 4 | Кнопка аварийного останова | Обрывает силовое питание привода и останавливает робота, когда эта кнопка нажата при непредвиденной ситуации. Силовое питание не отключается |
| 5 | Подключение первичного питания | Подключение кабеля первичного питания в контроллер |
| 6 | COM-порт | Подключение вспомогательных устройств и технологического оборудования |
| 7 | USB-порт | Подключение вспомогательных устройств |

4 Методы для останова робота.

Различают методы останова робота для режима обучения, автоматического режима и аварийного останова.

Для режима обучения:

- отпустите переключатель TRIGGER на пульте ручного управления;
- убедитесь, что робот полностью остановился, и нажмите клавишу HOLD на пульте ручного управления. В правом верхнем углу экрана пульта управления появится окно HOLD вместо окна RUN.

Для автоматического режима:

- чтобы остановить движение робота при выполнении программы в автоматическом режиме, нужно нажать клавишу HOLD на пульте ручного управления;
- чтобы закончить выполнение шага по программе и затем остановить движение робота, следует выбрать функцию Step Once нажатием в окне экрана пульта управления;
- чтобы закончить выполнение программы и затем остановить движение робота, нужно выбрать функцию Repeat Once нажатием в окне экрана пульта управления.

При аварийном останове: если робот работает неправильно и существует возможная опасность столкновения робота с предметами, получения травмы, то нужно немедленно нажать любую кнопку аварийного останова на пульте ручного управления или на операционной панели контроллера, чтобы отключить силовое питание привода.

Пульт ручного управления робота.

1 Описание пульта ручного управления.

На рисунке 1.3 представлены вид и краткое описание пульта ручного управления. Пульт ручного управления обеспечен аппаратными клавишами и переключателями, которые являются необходимыми для ручного управления роботом и данными редактирования и экраном для того, чтобы отображать различные виды данных. Расположение аппаратных клавиш показано на рисунке 1.4.

2 Аппаратные клавиши на пульте ручного управления (таблица 1.2).

3 Экран пульта ручного управления.

Пульт ручного управления обеспечен операционным дисплеем. Он состоит

из трех областей (А, В, С), как показано на рисунке 1.5.

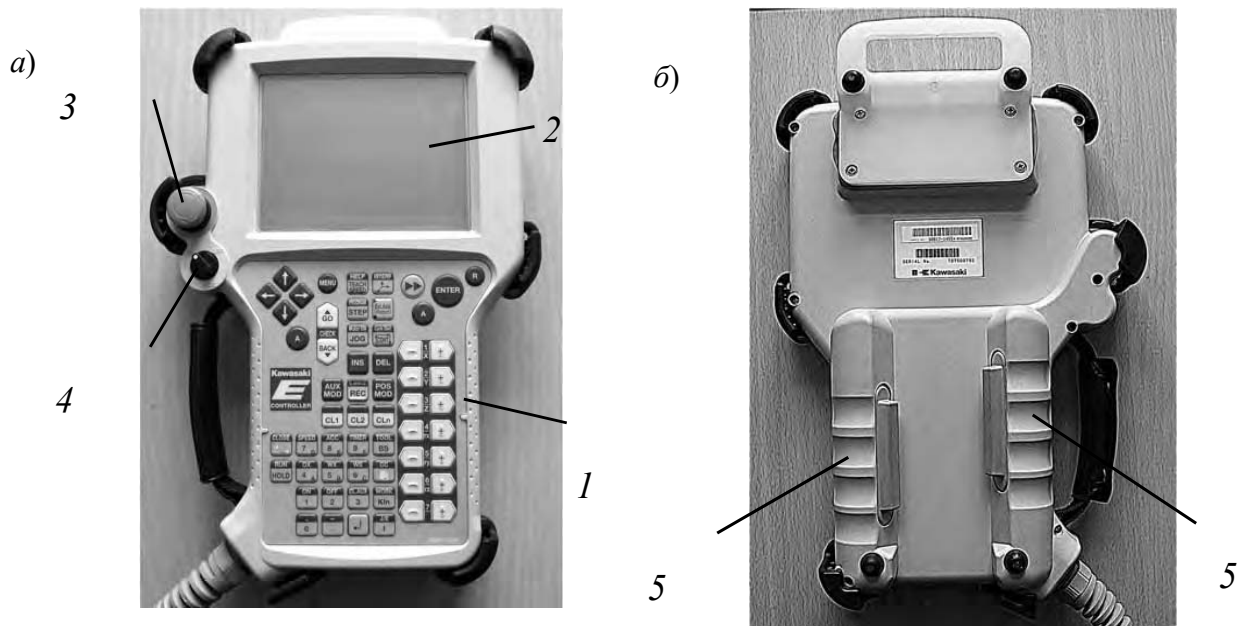
Область экрана А.

Эта область состоит из пяти экранов. Их функции приведены ниже.

*1 Функции в области **Program Comment**.*

Эта область показывает зарегистрированное имя программы и его комментарии. Пошаговое меню в этой области имеет шесть функций (рисунок 1.6). Чтобы вызвать пошаговое меню, нажмите А + PROGRAM на пульте ручного управления или область **Program Comment** непосредственно на экране.

В окно CALL PROGRAM можно ввести номер программы и выбрать программу, если имя программы состоит из чисел. Программы могут иметь имя до пяти символов.



а – со стороны дисплея; *б* – задняя сторона пульта со стороны триггера; *1* – панель с клавишами; *2* – экран; *3* – кнопка аварийного останова; *4* – переключатель Teach Lock; *5* – переключатель TRIGGER

Рисунок 1.3 – Вид пульта ручного управления



Рисунок 1.4 – Расположение аппаратных клавиш на пульте управления

Таблица 1.2 – Назначение основных аппаратных клавиш на пульте ручного управления и переключателей

| Тип клавиши, переключателя | Назначение клавиши, переключателя |
|-------------------------------------|---|
| MENU | Отображает пошаговое меню выбранной области дисплея. Нажимая A + MENU, активизируем пошаговое меню выбранной области дисплея |
| Стрелки вверх вниз, влево вправо | Передвигает курсор в вертикальном и горизонтальном направлениях между шагами, пунктами, строками. Одновременное нажатие «A + стрелка вверх или вниз» переводит курсор на предыдущий или последующий шаг в режиме обучения или редактирования |
| ENTER | Подтверждает выбор функции или пункта. Завершает ввод данных в активизированную строку выбранной области |
| R | Отмена последней операции. Закрывает пошаговое меню. Возврат в первоначальное состояние |
| A | «A» используется при выборе кнопок синего цвета (одновременное нажатие A+) |
| GO | Осуществляет пошаговое выполнение программных инструкций, переходя с выбранного шага к последующему |
| BACK | Осуществляет пошаговое выполнение программных инструкций, переходя с выбранного шага к предыдущему в проверочном режиме |
| TACH SPEED | Изменяет скорость движения манипулятора в режиме обучения |
| Оси координат | Задаёт систему координат: базовую (декартовую); относительную (движение по осям, связанным со схватом); угловую (движение отдельных суставов) |
| <u>PROGRAM</u> STEP | Нажимая A + PROGRAM, активизируем меню выбора программ. Нажимая STEP, активизируем меню пошаговой области |
| Ext. Axis (Robot) | Выбирает дополнительную ось (JT7) или еще одного робота, связанного с данным роботом |
| JOG | Медленно перемещает робота в режиме обучения или проверочном (контрольном режиме). Эффективно только в течение нажатия кнопки |
| CONT | Включает непрерывное или пошаговое выполнение программы |
| INS | Включает режим вставки программных шагов |
| DEL | Включает режим удаления программных шагов |
| AUX MOD | Включает режим редактирования дополнительной информации |
| POS MOD | Включает режим редактирования позиций |
| REC | Добавляет новый шаг следом за текущим шагом. Нажимая A + REC, записываем новый шаг в режиме обучения |
| CL1 | Переключает зажим 1 в положение ON/OFF |
| CL2 | Переключает зажим 2 в положение ON/OFF |
| CLn | Переключает n-й зажим в положение ON/OFF |
| Клавиши «+» и «-» | Двигает каждый сустав от JT1 до JT6 |
| ON | Нажимая A + ON, устанавливаем выбранный зажим в положение ON |
| OFF | Нажимая A + OFF, устанавливаем выбранный зажим в положение OFF |
| CLAUX | Нажимая A + CLAUX в режиме обучения, вызываем дополнительные пункты для зажимов |

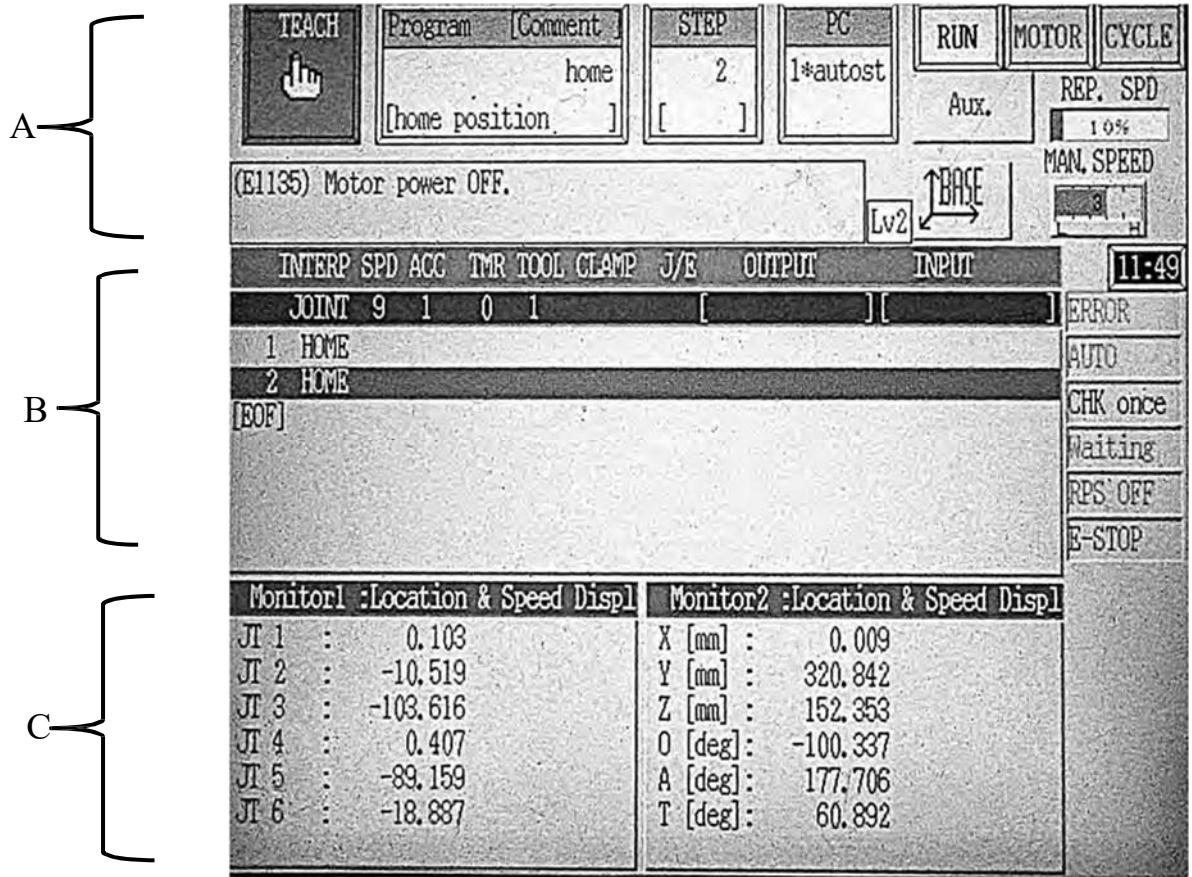


Рисунок 1.5 – Области дисплея

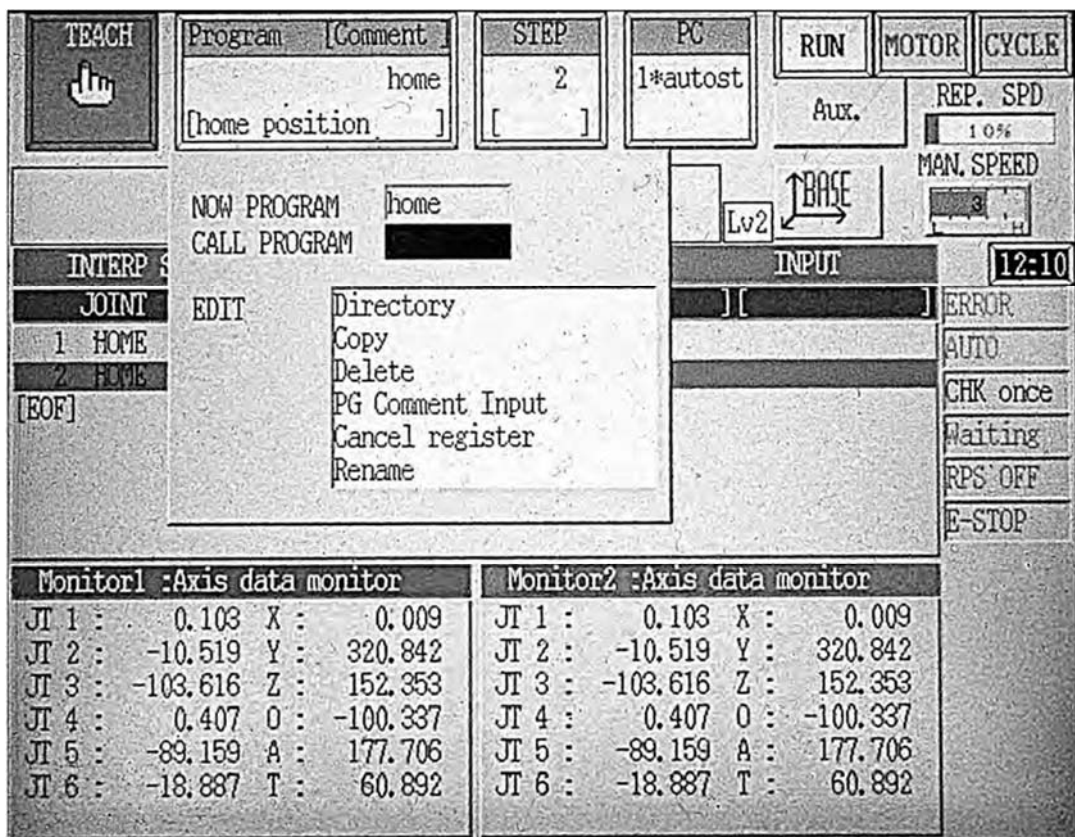


Рисунок 1.6 – Область PROGRAM COMMENT

Требуемая программа может быть выбрана из списка, если имя программы зарегистрировано. Нужно выбрать из пошагового меню функцию **Directory** (см. рисунок 1.6), после этого появится экран выбора программы (рисунок 1.7) и из списка нужно выбрать программу.

| Directory | | Program Name : 1 |
|-----------------|----------|------------------|
| PROGRAM NAME | STEP NUM | COMMENT |
| 1 | 0 | () |
| autostart.pc | 155 | () |
| demo | 31 | () |
| get_picture | 12 | () |
| home | 2 | home position) |
| kaw1 | 16 | home position) |
| kaw2 | 10 | home position) |
| kaw3 | 0 | () |
| 899 | 5 | () |
| primer1 | 20 | () |
| prog1 | 2 | () |
| prog2 | 6 | () |
| prog3 | 17 | () |
| tcpcomm_close | 17 | () |
| tcpcomm_connect | 50 | () |
| tcpcomm_end | 1 | () |
| tcpcomm_rcv | 38 | () |
| tcpcomm_send | 41 | () |

Input Next Page

Рисунок 1.7 – Экран Directory выбора программы

Выбранная программа может быть удалена с помощью функции **Delete** (см. рисунок 1.6).

Комментарии могут быть введены в специальную область выбранной программы с помощью функции **PG Comment Input**.

С помощью функции **Copy** можно скопировать содержание выбранной программы, изменив имя программы. Эту функцию можно использовать для создания новой программы.

С помощью функции **Cancel Register** программа может быть удалена из области **Program Comment**.

2 Функции в *Step*-области.

Эта область показывает номер шага программы. Пошаговое меню в данной области имеет пять функций (рисунок 1.8). Чтобы показать пошаговое меню, нужно нажать напрямую на экране пошаговую область.

На рисунке 1.8 в открытом пошаговом меню показан текущий шаг загруженной программы в поле NOW STEP – 16. Имя программы «kaw1» отображается в окне **Program Comment**. В поле вызова шага CALL STEP можно ввести номер шага программы, нажать кнопку ENTER на пульте, и этот шаг будет выбран (рисунок 1.9, выбран шаг 4).

Последний шаг в программе можно выбрать с помощью функции **Bottom** (рисунок 1.8, выбран последний шаг – 16).

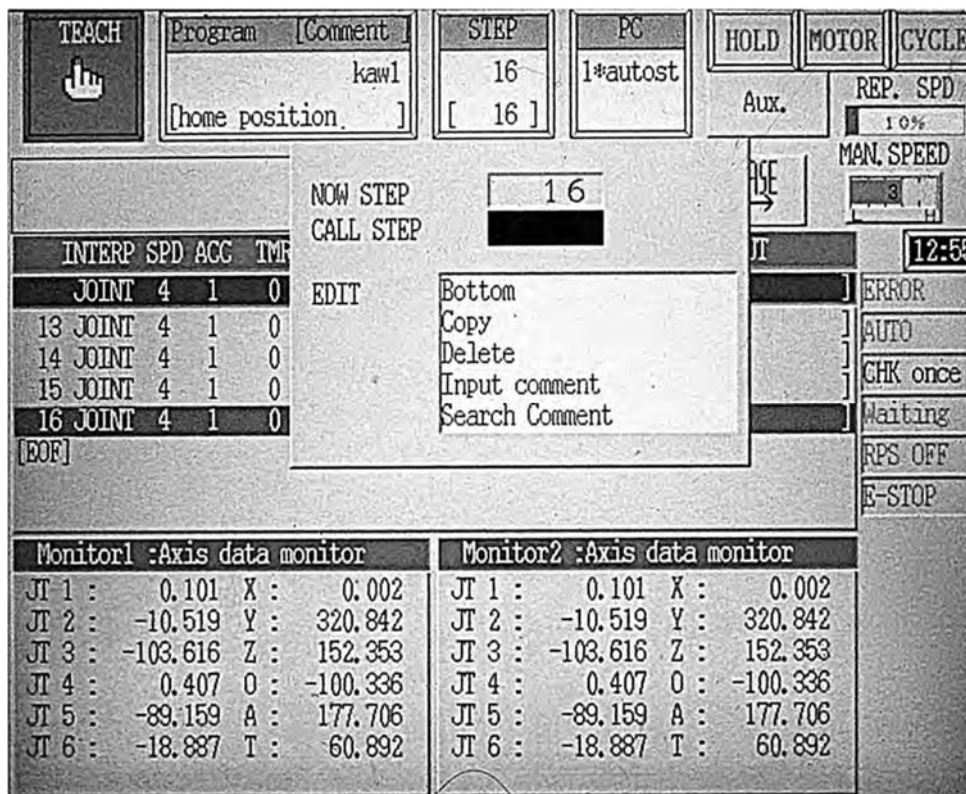


Рисунок 1.8 – Step-область экрана

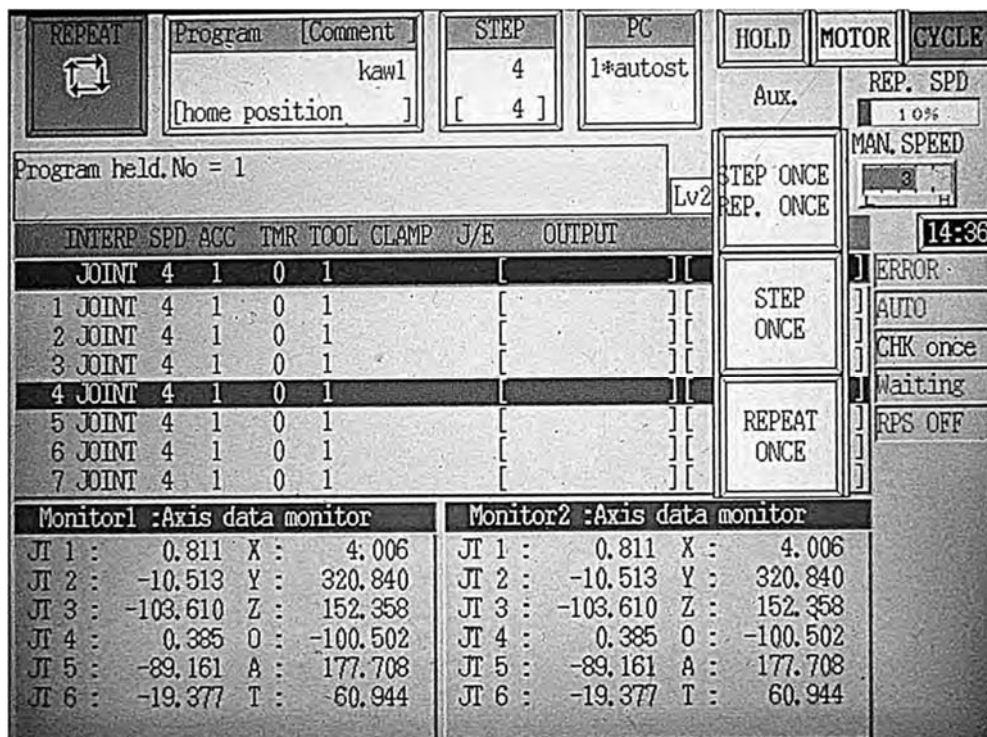


Рисунок 1.9 – Окна отработки программы в автоматическом режиме (пошаговая отработка)

3 Область системных сообщений.

Эта область показывает сообщения об ошибках и различные виды сообщений, которые возникают в течение работы робота. На рисунке 1.5 в окне

системных сообщений указаны коды ошибки (E1135) и сообщение «Motor power OFF» (электропитание двигателя выключено).

4 Область состояния.

Эта область расположена справа от области РС вверху рабочего экрана на пульте ручного управления и состоит из окон и режимов (см. рисунок 1.5).

1 Окно MAN. SPEED (manual speed) устанавливает скорость движения робота в режиме обучения (ручном режиме). Переключать скорость можно, нажав клавишу TEACH SPEED на пульте управления или касанием на экране в области окна.

2 Окно REP. SPD (Repeat speed) отображает скорость, когда робот перемещается в автоматическом режиме.

3 Три режима движения робота в режиме обучения:

- BASE – базовая система координат;
- TOOL – инструментальная система координат;
- JOINT – режим посуставного движения.

Нажимая клавишу с осями координат на пульте управления или соответствующее окно в области состояния, устанавливаем необходимую систему координат.

4 Окно MOTOR активно, когда силовое питание включено, и неактивно, когда силовое питание выключено.

5 Окно HOLD появляется, когда происходит останов робота, окно RUN – когда робот в состоянии выполнения.

6 Пошаговая отработка программы в режиме обучения. При нажатии клавиши CONT на пульте управления на экране справа в окне отобразится CHK once. Выбирая шаг программы, нажав клавиши A + стрелки вверх и вниз на пульте управления, затем, нажав и удерживая триггер и клавишу GO, выполняем программу в пределах одного шага. Далее выбираем и выполняем второй шаг и т. д.

7 Отработка программы в автоматическом режиме. При касании на экране в окнах функции STEP CONT и REPEAT ONCE загруженная программа выполняется только один раз или непрерывно.

8 Пошаговая отработка программы в автоматическом режиме. Выбрав окно STEP ONCE на экране и нажимая клавишу GO на пульте управления, выполняем программу пошагово (см. рисунок 9).

Область экрана В.

Левая сторона этой области содержит пошаговое меню (рисунок 1.10):

- **Teach** – экран обучения. Используется для обучения или редактирования программы робота в блочном программировании (на рисунке 1.10 экран открыт);
- **Aux Function** – экран вспомогательных функций. Используется для установки различных видов данных робота;
- **I/F Panel** – экран интерфейсной панели. Обеспечивает выключатели и функции, необходимые для операций роботов и периферийного оборудования;
- **Keyboard** – экран клавиатуры. Используется для ввода символов имени программы, комментариев и т. д.;
- **Monitor1** и **Monitor2** – экраны контроля открываются в области С.

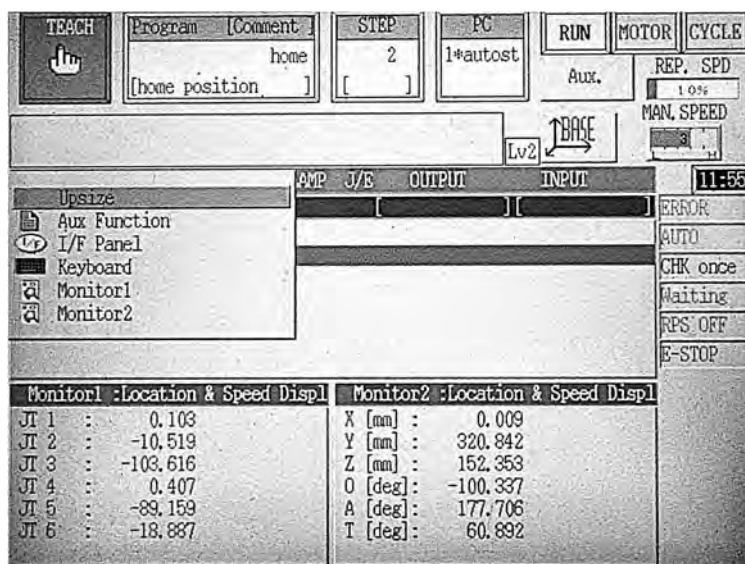


Рисунок 1.10 – Область экрана В

Чтобы отобразить пошаговое меню, нужно нажать клавиши A + MENU на пульте управления или окно области В непосредственно. Выбор Upsize увеличивает и отображает содержание областей В и С.

Область экрана С.

Эта область содержит два вида экранов, каждый из которых становится доступным из пошагового меню.

1 Экран контроля суставов. Этот экран отображает восемь видов информации, включая текущее положение робота, командные значения и т. д.

2 Экран контроля сигнала. Этот экран отображает предварительно установленные имена входов-выходов и контролирует состояние каждого сигнала.

Задания к практическому занятию

1 Изучить панель контроллера Kawasaki FD70. Изучить функциональное назначение переключателей на панели, внешних входов/выходов.

2 Изучить назначение аппаратных клавиш на пульте ручного управления.

3 Изучить назначение и функционирование областей экрана А, В, С операционного дисплея пульта ручного управления.

4 Изучить методы включения-выключения контроллера, силового питания робота.

После изучения структуры и компонентов учебного роботизированного комплекса студенты под руководством преподавателя должны практически выполнить:

- включить контроллер и силовое питание;
- на пульте управления загрузить одну из программ из списка окна Directory, выбрать отдельные шаги программы;
- создать имя новой программы;

- установить различную скорость перемещения робота в ручном режиме;
- переключиться в каждый из трех режимов движения робота в режиме обучения;
- открыть меню в каждой области экрана А, В, С и изучить содержание функций в пошаговом меню и в открывающихся окнах;
- выключить силовое питание и контроллер.

Контрольные вопросы к практическому занятию № 1

- 1 Какие функции осуществляет внешняя передняя панель контроллера или системы управления (СУ) робота?
- 2 Назовите способы останова робота.
- 3 Что расположено на пульте ручного управления робота и для чего он предназначен?
- 4 Из каких областей состоит операционный дисплей пульта ручного управления?

Контрольные вопросы к практическому занятию № 2

- 1 Что показывает область Program Comment на экране пульта ручного управления робота?
- 2 Что показывает область системных сообщений на экране пульта ручного управления робота?
- 3 Что показывает область экрана В на пульте ручного управления робота?
- 4 Что показывает область экрана С на пульте ручного управления робота?

2 Практическое занятие № 3. Позиционные и кинематические задачи управления роботами. Практическое занятие № 4. Изучение методов ручного управления роботом

Система координат робота.

1 Имена осей.

Робот имеет шесть осей, как показано на рисунке 2.1. Эти оси обозначаются JT1–JT6.

2 Ручное управление седьмой осью.

Седьмая ось – дополнительная ось, которая добавляется и может быть продольной осью, сварочные клещи с серводвигателем и т. д. Ручное управление, как и при действии с шестью осями.

3 Ручное управление восьмьюми-восемнадцатью осями.

Этот контроллер может управлять максимумом 18 осями. Ручное управление этими осями, как и для седьмой оси. Для того чтобы использовать дополнительные оси, или 8–14, или 15–18, нажимают EXT AXIS (Robot).

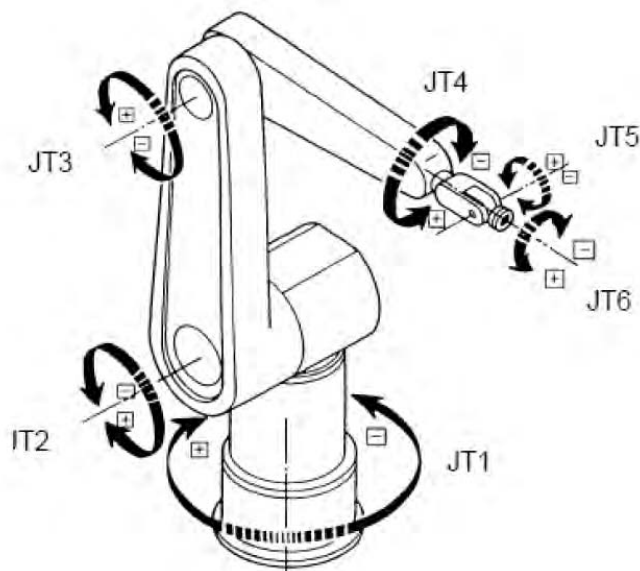


Рисунок 2.1 – Схема осей робота

Режимы ручного управления робота.

1 Режим суставных координат.

Нажав клавишу с координатными осями на пульте ручного управления или непосредственно в окне области состояния дисплея, можно изменить изображение текущего режима на режим суставных координат Joint. Когда этот режим выбран, оси робота могут перемещаться индивидуально, как показано на рисунке 2.1. Когда нажимается несколько клавиш осей одновременно, оси робота могут перемещаться в комбинации.

2 Режим основных координат.

Изменим изображение режима на режим основных координат Base с помощью пульта ручного управления или нажатием области в окне дисплея.

Когда этот режим выбран, оси робота могут перемещаться, базирясь на основной системе координат. Когда нажато несколько клавиш одновременно, оси робота могут перемещаться в комбинации (рисунок 2.2).

3 Режим инструментальных координат.

Изменим изображение режима на режим инструментальных координат Tool с помощью пульта ручного управления или нажатием области в окне дисплея. Когда этот режим выбран, оси робота могут перемещаться, базирясь на системе координат инструмента.

Система координат инструмента использует набор координат, который определяет положение инструмента в пространстве. Установка инструментальной системы координат меняется каждый раз, когда изменяются позиция и положение робота.

Значения инструментальной системы координат должны быть изменены, когда используется новый инструмент, имеющий другую форму.

На рисунке 2.3 показано, как изменяется положение системы координат в зависимости от положения в пространстве робота. При горизонтальном положении робот имеет инструментальную систему координат X_1, Y_1, Z_1 . При перемещении робота вниз положение инструментальной системы координат изменится на X_2, Y_2, Z_2 .

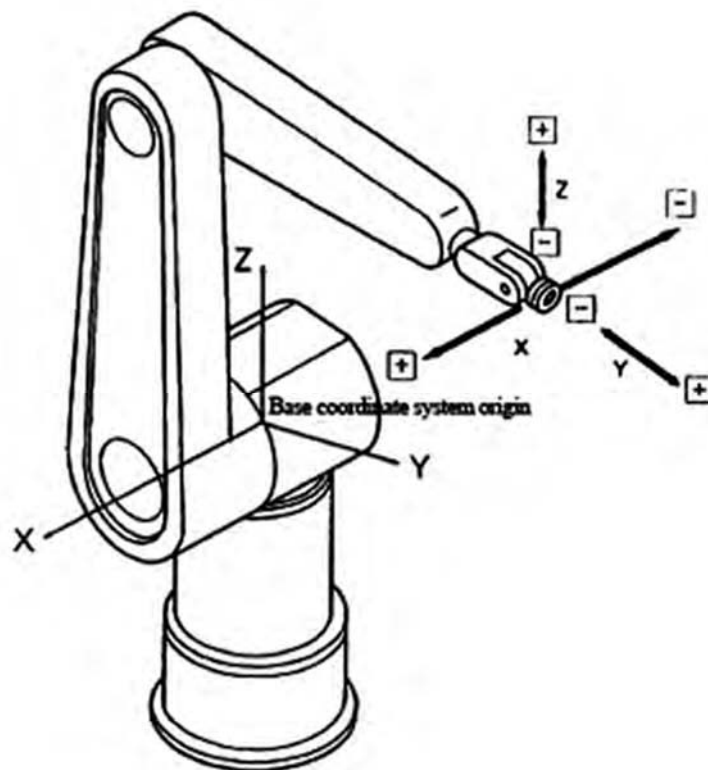


Рисунок 2.2 – Основная базовая и инструментальная системы координат

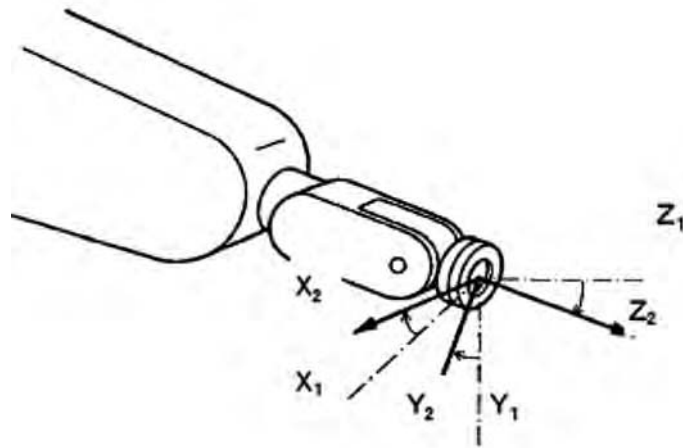


Рисунок 2.3 – Положение робота, перемещенного вниз относительно сустава JT3, в режиме инструментальных координат

Когда положение запястья меняется, положение инструментальной системы координат также изменяется.

Задания к практическому занятию

После изучения режимов ручного управления робота студенты под руководством преподавателя выполняют операции по управлению каждой из шести осей.

Последовательность выполнения операций по управлению перемещений робота в ручном режиме.

1 Включите переключатель силового питания контроллера и убедитесь, что лампа силового питания включилась.

2 На панели контроллера поверните переключатель TEACH/REPEAT в положение TEACH (горизонтальное положение).

3 Включите в положение ON переключатель TEACH LOCK на пульте ручного управления.

4 Нажмите клавишу с координатными осями на пульте ручного управления или непосредственно в окне области состояния дисплея, чтобы установить режим операции Joint, Base или Tool.

5 Установите скорость операции, нажимая клавишу TEACH SPEED на пульте или окно в области состояния на дисплее.

6 Когда шаги 1–5 закончены, включите силовое питание привода (A + Motor ON).

7 Поверните переключатель HOLD/RUN в позицию RUN.

8 Пока нажат переключатель TRIGGER на пульте ручного управления, можно перемещать робот, нажимая клавиши осей «+» и «-» от 1 до 6. Робот продолжит двигаться, пока клавиши нажаты.

9 При отпуске клавиш «+» и «-» или переключателя TRIGGER на пульте ручного управления робот останавливается.

После завершения ручного управления установите в положение OFF

переключатель TEACH LOCK на пульте ручного управления.

Контрольные вопросы к практическому занятию № 3

- 1 Сколько систем координат и осей имеет робот?
- 2 Для чего предназначена седьмая ось робота?
- 3 Как осуществляется ручное управление восьмьюми-восемнадцатыми осями?
- 4 Как переместить схват робота одновременно по нескольким осям в ручном режиме?

Контрольные вопросы к практическому занятию № 4

- 1 Охарактеризуйте режим суставных координат.
- 2 Охарактеризуйте режим основных координат.
- 3 Охарактеризуйте режим инструментальных координат.
- 4 В каких случаях изменяют значения инструментальной системы координат?

3 Практическое занятие № 5. Управление схватом манипуляционного робота по вектору скорости. Практическое занятие № 6. Блочное программирование робота

Метод блочного программирования заключается в обучении робота и редактировании программы с помощью пульта ручного управления.

Подготовка для обучения робота.

1 Установка переключателя TEACH LOCK.

В начале обучения нужно установить переключатель TEACH LOCK на пульте ручного управления в положение ON и переключатель на контроллере TEACH/REPEAT в положение TEACH.

Робот не будет двигаться, если переключатель на контроллере TEACH/REPEAT установлен в автоматический режим REPEAT. Кроме того, если переключатель на пульте ручного управления TEACH LOCK находится в состоянии OFF, робот не может перемещаться вручную в режиме обучения.

2 Конфигурация экрана обучения.

При обучении программ в блочном программировании используется экран обучения.

Чтобы отобразить экран обучения, нужно активизировать область экрана В и нажать клавиши пульта управления А + MENU или окно области В, чтобы отобразить пошаговое меню, затем выбрать пункт Teach из меню.

Конфигурация экрана обучения следующая (рисунок 3.1).

Главную (самую высокую) строку называют строкой заголовка, и пункты, необходимые для блочного программирования, отображаются в этой строке.

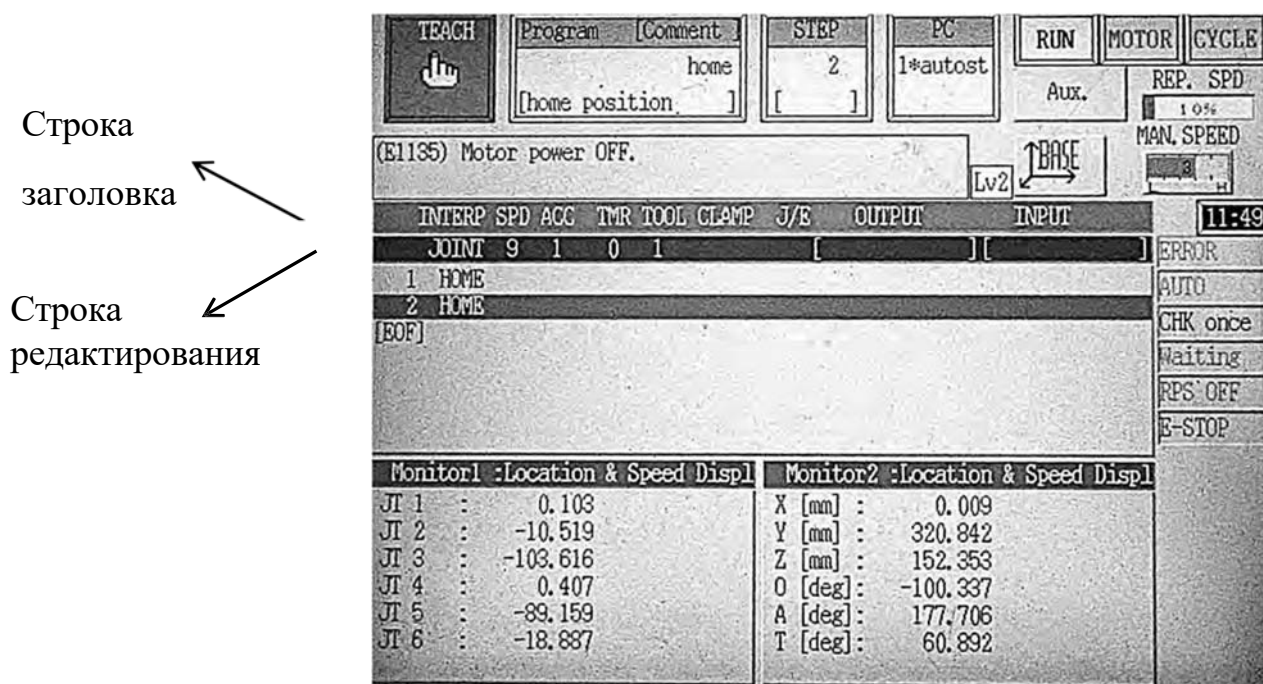


Рисунок 3.1 – Конфигурация экрана обучения

Строка редактирования расположена ниже строки заголовка. Эта строка используется, чтобы редактировать содержание каждого шага.

Для переключения из одного столбца (элемента) строки заголовка в другой нужно переместить курсор при помощи клавиш со стрелками «влево-вправо» пульта управления. Изменения значений вспомогательных данных (элементов) производятся с помощью клавиш со стрелками «вверх-вниз» на пульте управления или вводом чисел напрямую.

Содержание каждого элемента в строке заголовка показано в таблице 3.1.

Таблица 3.1 – Содержание элемента в строке заголовка экрана обучения

| Название элемента (вспомогательных данных) | Содержание элемента |
|---|--|
| Interpolation | Определяет способ перемещения манипулятора к каждой обученной точке непрерывно. Например, выбор линейной интерполяции сделает траекторию между точками прямолинейной |
| Speed | Задаёт скорость движения в обученную точку |
| Accuracy | Определяет порядок, в котором робот приближается и считает, что достиг точки |
| Timer | Определяет время ожидания в обученной точке |
| Tool | Определяет номер рабочего инструмента, смонтированного на работе |
| Clamp | Задаёт ON/OFF состояние схвата при взятии детали |
| J/E | Задаёт переключение программ при помощи внешнего сигнала |
| O/X | Задаёт выходной сигнал от робота к периферийному оборудованию |
| W/X | Задаёт входной сигнал к робота от периферийного оборудования |

Операции обучения робота.

Обучение робота методом блочного программирования выполняется на экране обучения с помощью пульта ручного управления.

Пример обучения робота.

В качестве примера рассмотрим, как обучить робота, который будет перемещаться по заданной траектории к каждой из четырех точек, как показано на рисунке 3.2.

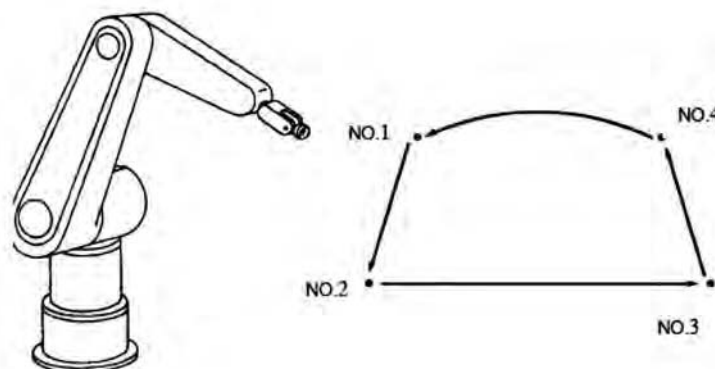


Рисунок 3.2 – Обучение робота по заданной траектории

Сначала нужно создать имя новой программы или загрузить имеющуюся программу, которая должна быть отредактирована.

Задание на обучение робота показано в таблице 3.2.

Таблица 3.2 – Содержание процедуры обучения робота

| Обучаемый шаг программы | Содержимое обучения |
|-------------------------|--|
| Шаг 1 | Перемещение в начальную точку робота N0.1 |
| Шаг 2 | Движение из точки N0.1 в точку N0.2 с медленной скоростью по линейной траектории, с высокой точностью, с небольшой задержкой |
| Шаг 3 | Движение от точки N0.2 в точку N0.3 с медленной скоростью по линейной траектории со сменой инструмента |
| Шаг 4 | Движение из точки N0.3 в точку N0.4 с более высокой скоростью по прямолинейной траектории |
| Шаг 5 | Движение из точки N0.4 в точку N0.1 с более высокой скоростью по траектории Joint |

Последовательность выполнения задания.

1 Перемещаем робот в точку N0.1 для обучения шага 1 при помощи клавиш «+» «-» и удерживая переключатель TRIGGER. Нажимаем клавишу REC на клавиатуре пульта ручного управления (ПРУ) для того, чтобы обучить позицию. Скорость, точность перемещения и вид траектории не устанавливаем, они остаются текущими.

2 Перемещаем робота из точки N0.1 в точку N0.2 для обучения шага 2 при помощи клавиш «+» «-» и удерживая переключатель TRIGGER. Устанавливаем линейную интерполяцию, скорость 3, точность 7, таймер 1 следующим образом.

Для установки линейной интерполяции перемещаем курсор в заглавной строке на колонку INTERP (см. рисунок 3.1), нажимая клавиши (ПРУ) A + INTERP или при помощи клавиш со стрелками «вправо-влево». Нажимая клавишу (ПРУ) со стрелкой вверх или нажимая на экране область INTERP, изменяем режимы, установленные в строке редактирования на режим LINEAR.

Для установки скорости движения робота перемещаем курсор в заглавной строке на колонку SPD, нажимая кнопки A + SPEED или при помощи клавиш со стрелками «вправо-влево». Устанавливаем скорость 3 с помощью клавиш со стрелками «вверх-вниз» или нажимая на экране область SPD.

Для установки точности движения робота перемещаем курсор в заглавной строке на колонку Acc (точность), нажимая клавиши A + ACC или при помощи клавиш со стрелками «вправо-влево». Устанавливаем точность 7 с помощью клавиш со стрелками «вверх-вниз» или нажимая на экране область ACC.

Для установки таймера перемещаем курсор в заглавной строке на колонку TMR, нажимая клавиши A + TIMER или при помощи клавиш со стрелками «вправо-влево». Устанавливаем таймер 1 с помощью клавиш со стрелками «вверх-вниз» или нажимая на экране область TMR. Нажимаем клавишу REC на клавиатуре ПРУ для того, чтобы обучить позицию.

3 Перемещаем робота из точки N0.2 в точку N0.3 для обучения шага 3 при помощи клавиш «+» «-» и удерживая переключатель TRIGGER. Не изменяем линейную интерполяцию, скорость 3 и точность 7. Таймер не устанавливаем.

Меняем инструмент, устанавливаем инструмент 2. Для установки инструментальной системы координат перемещаем курсор в заглавной строке на колонку TOOL, как в предыдущем шаге. Изменяем инструментальную систему координат в строке редактирования, нажимая клавиши A + TOOL или при помощи клавиш со стрелками «вправо-влево». Устанавливаем номер инструмента 2. Нажимаем клавишу REC на клавиатуре ПРУ для того, чтобы обучить позицию.

4 Перемещаем робота в точку N0.4 для обучения шага 4. Устанавливаем линейную интерполяцию, скорость 6, точность 3, как в предыдущих шагах. Нажимаем клавишу REC на клавиатуре ПРУ для того, чтобы обучить позицию.

5 Перемещаем робота в точку N0.5 для обучения шага 5. Устанавливаем суставную интерполяцию JOINT в колонке INTERP, скорость и точность не изменяем. Режим интерполяции устанавливаем так же, как во втором шаге. Нажимаем клавишу REC на клавиатуре ПРУ для того, чтобы обучить позицию.

Операции обучения для данной программы завершены.

Каждый шаг программы должен заканчиваться нажатием клавиши REC для записи в память робота.

Проверка работы программы.

Для того чтобы подтвердить воспроизведение обученной программы, используются клавиши пульта управления GO и BACK в режиме проверки.

1 Выбираем программу, которая должна быть проверена.

2 Можно начать проверку с первого шага или с любого выбранного шага; для этого устанавливаем шаг программы.

3 Переключаемся в режим обучения и переключаем TEACH LOCK на пульте ручного управления в состояние ON. Устанавливаем TEACH/REPEAT в положение TEACH.

Нажатие клавиши CONT на пульте управления позволяет изменять метод проверки, выполнение одного приращения шага или непрерывное выполнение. Метод проверки отображен в области состояния.

4 Устанавливаем скорость проверки.

5 После включения силового питания привода нужно нажать клавишу RUN на пульте ручного управления и переключиться на окно RUN на дисплее для того, чтобы перемещать робота при помощи пульта ручного управления.

6 Нажимаем переключатель TRIGGER на пульте ручного управления и клавишу GO, чтобы переместить робота к выбранному шагу.

7 Когда проверяется приращение в одном шаге ONCE, робот останавливается по завершении движения.

8 При непрерывной проверке шагов CONT робот перемещается непрерывно в следующий шаг в то время, как GO нажато.

Редактирование данных программы.

Есть четыре основные процедуры для редактирования обученных данных

программы.

- 1 Перезапись позиции.
- 2 Перезапись вспомогательных данных.
- 3 Вставка шагов.
- 4 Удаление шагов.

Перезапись позиции выполняется в следующей последовательности.

1 Устанавливаем курсор на шаг, который нуждается в редактировании, нажимая клавиши на пульте управления А и клавиши со стрелками «вверх-вниз». Движение к шагу видно по перемещению строки зеленого цвета.

2 Нажимаем клавиши на пульте управления А + POS/MOD, цвет строки шага изменяется и в левой части строки редактирования появляется изображение POS.M.

3 Подводим манипулятор к редактируемой позиции, нажимая клавиши на пульте управления «+», «-».

4 Нажимаем клавишу REC, тем самым регистрируем новую позицию шага. Редактирование закрывается нажатием клавиши А + POS/MOD.

Перезапись вспомогательных данных выполняется с помощью клавиши пульта управления А + AUX/MOD, вставка шагов – с помощью клавиши А + INS, удаление шагов – с помощью клавиши А + DEL. Последовательность выполнения этих функций аналогична перезаписи позиций.

Запуск выбранной программы в автоматическом режиме.

Рассмотрим основные методы для запуска робота в автоматическом режиме с помощью пульта ручного управления. Эти методы называют операциями повтора.

1 Загружаем программу, которую хотим выполнить.

2 Устанавливаем главный переключатель контроллера TEACH/REPEAT в положение REPEAT.

3 Устанавливаем переключатель пульта ручного управления TEACH LOCK в положение OFF.

4 Устанавливаем необходимые состояния выполнения программы (непрерывно, выполнение только один раз, пошаговое выполнение).

5 Нажимаем клавиши А + Motor ON на пульте управления (включаем привод робота). Кнопка MOTOR на дисплее станет активной.

6 Нажимаем клавиши А + RUN, должна отобразиться на дисплее кнопка RUN (даем разрешение на выполнение программы).

7 Нажимаем Cycle Start на контроллере. Кнопка CYCLE на дисплее станет активной. Робот начинает выполнение программы.

Когда переключатель TEACH LOCK на пульте в состоянии ON, автоматический режим невозможен.

Есть два способа остановки робота во время автоматического режима: прерывание программы или завершение выполнения программы.

Способы прерывания программы:

- 1) нажать клавишу HOLD на пульте ручного управления;
- 2) когда робот заканчивает шаг движения полностью, установить

переключатель пульта управления TEACH LOCK в положение ON, что отключит силовое питание привода;

3) когда робот заканчивает шаг движения полностью, нажать любую аварийную кнопку для выключения силового питания привода.

Для завершения выполнения программы нужно установить состояние REPEAT CONT/ONCE в положение ONCE в окне дисплея.

Основные технические характеристики промышленного робота представлены в таблице 3.3.

Таблица 3.3 – Технические характеристики манипулятора Kawasaki FS003N

| Параметр | Техническая характеристика | | |
|---|--|-----------------|--------------------------|
| 1 Модель | FS003N | | |
| 2 Тип | Сочлененный (антропоморфный) робот | | |
| 3 Степеней подвижности | 6 | | |
| 4 Угол поворота и максимальная скорость | Поворот руки (JT1) | 320° | 360°/с |
| | Наклон руки (JT2) | 210° | 250°/с |
| | Рука вверх-вниз (JT3) | 270° | 225°/с |
| | Поворот кисти (JT4) | 720° | 540°/с |
| | Изгиб кисти (JT5) | 270° | 225°/с |
| | Вращение кисти (JT6) | 720° | 5440°/с |
| 5 Повторяемость | ± 0,05 мм | | |
| 6 Грузоподъемность | 3 кг | | |
| 7 Максимальная скорость | Макс. 6000 мм/с в прямолинейном движении | | |
| 8 Допустимая нагрузка кисти | | Крутящий момент | Момент инерции |
| | JT4 | 5,8 Н · м | 0,12 кг · м ² |
| | JT5 | 5,8 Н · м | 0,12 кг · м ² |
| | JT6 | 2,9 Н · м | 0,03 кг · м ² |
| 9 Масса | 20 кг | | |

Задания к практическому занятию

1 Выполнить задание, представленное в таблице 3.2, следуя описанным инструкциям по обучению робота.

2 Выполнить индивидуальное задание по блочному программированию робота. Задание выдается преподавателем.

Контрольные вопросы к практическому занятию № 5

1 Назовите основные процедуры редактирования данных программы для перемещения робота.

2 Перечислите методы для запуска робота в автоматическом режиме.

3 Назовите основные технические характеристики промышленного робота.

4 Назовите основные технические характеристики контроллера промышленного робота.

Контрольные вопросы к практическому занятию № 6

- 1 В чем заключается метод блочного программирования робота?
- 2 Перечислите основные этапы подготовки для обучения робота.
- 3 Какова последовательность обучения робота?
- 4 Какие клавиши пульта используют для проверки работы программы управления роботом?

4 Практическое занятие № 7. Изучение языка программирования AS робота Kawasaki. Разработка программ

Основы программирования робота на языке AS.

Роботы Kawasaki управляются системной программой, называемой AS. В AS-системе можно задавать команды или выполнять программы, используя AS-язык программирования. AS-система записана в энергонезависимой памяти контроллера, в оперативном запоминающем устройстве (ОЗУ). Когда управляющее питание включается, AS-система запускается и ожидает ввода команд для выполнения.

Язык программирования AS разделяется на два вида: мониторные команды и программные инструкции.

Мониторные команды используются для написания, редактирования, выполнения программ и единичных команд. Они вводятся после знака (>), появляющегося в начале строки, и выполняются после нажатия клавиши **Enter** немедленно. Некоторые мониторные команды применяются внутри программы как программные инструкции.

Программные инструкции используются для создания последовательности движения робота, для контроля и управления внешними сигналами в программах.

Программа есть совокупность программных инструкций.

В AS-системе возможны два способа задания координатной точки в трехмерном пространстве:

1) последовательностью угловых смещений каждой из осей суставов относительно выбранного начального положения (угловая координата);

2) последовательностью линейных и угловых смещений относительно начала базовой системы координат (декартова координата).

При запоминании точки в угловых координатах необходимо перед буквой поставить знак #.

Пример записи угловой координаты:

| | JT1 | JT2 | JT3 | JT4 | JT5 | JT6 |
|-------|-----|-----|-----|-----|-----|-----|
| #pose | 0 | 33 | -15 | 0 | -40 | 30 |

JT1–JT6 – оси суставов (шарниров) робота.

Пример записи декартовой координаты:

| | X | Y | Z | O | A | T |
|------|---|------|-----|---|---|---|
| pose | 0 | 1434 | 300 | 0 | 0 | 0 |

Координата задается значениями X, Y, Z для основных координат и положением O, A, T углов Эйлера. Имя декартовой координаты начинается с буквы.

Каждая строка (шаг) программного языка AS выражена в формате

номер шага метка программные инструкции; комментарии

Номер шага назначается для каждой строки программы. Шаги нумеруются последовательно начиная с 1 (при вводе с терминала автоматически нумеруются и перенумеровываются каждый раз, когда строки вставляются или удаляются).

Метка появляется при записи или редактировании программы с терминала в виде значка «?». При записи программы вне терминала метка не вводится.

Программные инструкции – это команды для перемещения робота и другие внешние команды.

Точка с запятой (;) указывает, что вся информация направо от точки с запятой (;) – это *комментарии*.

Глобальные переменные – это переменные, которые определены и сохраняются с этим значением в памяти. Они могут использоваться в любой программе.

В отличие от глобальных переменных, *локальные переменные* переопределяются каждый раз, когда программа выполняется, и не сохраняются в памяти. Переменная с точкой в начале имени считается локальной переменной (.pose). Локальные переменные полезны в случаях, когда несколько программ используют то же имя переменной.

Программные инструкции (команды) движения робота.

1 Формат программных инструкций:

Ключевое слово параметры;

JMOVE имя переменной позиции, номер зажима.

Параметр, указанный после запятой, может быть опущен. Всегда водится пробел между ключевым словом и параметрами.

Основные программные инструкции (команды) для создания последовательности движения робота:

JMOVE – перемещается в интерполированном угловом движении;

LMOVE – перемещается в интерполированном прямолинейном движении;

JAPPRO – приближается к координате в интерполированном угловом движении;

LAPPRO – приближается к координате в интерполированном прямолинейном движении;

JDEPART – отходит от текущей позиции в интерполированном прямолинейном движении;

LDEPART – отходит от текущей позиции в интерполированном угловом движении;

HOME – перемещается в начальную позицию; **C1MOVE** – перемещается с круговой интерполяцией; **C2MOVE** – перемещается с круговой интерполяцией.

2 Описание отдельных команд.

Команды: **JAPPRO**, **LAPPRO**.

Форматы команд:

JAPPRO имя переменной позиции, расстояние;

LAPPRO имя переменной позиции, расстояние.

Назначение команд. Робот двигается по оси Z инструментальной системы координат, не доходя определенного расстояния до обученной точки. Движение происходит: JAPPRO – по криволинейной траектории, LAPPRO – по прямолинейной траектории. Точка задается в декартовых или угловых координатах.

Параметр имя переменной позиции. Задается точка назначения в декартовых или угловых координатах.

Параметр расстояние. Задается расстояние между обученной точкой и положением в пространстве по оси Z , не доходя до обученной точки на заданное расстояние. Если расстояние задано положительным числом, робот не доходит до точки по оси Z выше точки, если отрицательным – ниже точки.

Пример – JAPPRO точка1, 200; криволинейное перемещение, не доходя до точки 1 на 200 мм по оси Z выше точки.

Команды: **JDEPART, LDEPART.**

Форматы команд:

JDEPART расстояние;

LDEPART расстояние.

Назначение команд. Манипулятор отходит от данной точки по оси Z инструментальной системы координат на определенное расстояние. Движение может быть криволинейным **JDEPART** или прямолинейным **LDEPART**.

Параметр расстояние. Задается в миллиметрах между текущей позицией и позицией, расположенной на оси Z , куда робот должен выйти. Если задано положительное значение, робот движется вверх по оси Z , если отрицательное – вниз по оси Z .

Пример – LDEPART-100; перемещение по оси Z на расстояние 100 мм вниз.

Программные инструкции (команды) двоичных сигналов.

Команда: **SIGNAL.**

Формат команды: **SIGNAL сигнальный номер.**

Назначение команды. Программируются только выходные сигналы. Включаются (ON), выключаются (OFF) заданные внешние или внутренние сигналы.

Параметр – сигнальный номер. Задается номер внешнего выходного сигнала или внутреннего сигнала. Сигнальный номер определен, если сигнал задан как внешний выходной сигнал числами 1–32, внешние входные сигналы не задаются. Если сигнальное число положительное, сигнал устанавливается в ON, если отрицательное – в OFF.

Пример – SIGNAL-1, 4 – выключает *первый* выход, включает *четвертый* выход на контроллере.

Внутренние сигнальные номера задаются числами 2001–2256.

Команда: **SWAIT.**

Форматы команды: **SWAIT сигнальный номер.**

Назначение команды. Программируются в основном входные сигналы. Ожидает, пока внешний или внутренний сигнал приобретет заданное состояние.

Параметр сигнальный номер. Задается сигнальный номер для контроля состояния. Отрицательные числа указывают, что состояние удовлетворяется, когда сигналы выключены. Командой SWAIT сигналы задаются следующими сигнальными номерами:

- внешние выходные сигнальные номера 1–32;
- внешние входные сигнальные номера 1001–1032;
- внутренние сигнальные номера 2001–2256.

Пример SWAIT-1007; внешний входной сигнал, ожидание пока входной сигнал не выключится.

Программные инструкции (команды) управления скоростью перемещения звеньев робота.

Команда SPEED задает скорость движения робота (программную скорость).

Формат команды:

SPEED speed, rotational speed, ALWAYS.

Параметр speed – скорость. Выбирается программная скорость движения робота, задается в процентах от 0,01 до 100 %, также в миллиметрах в секунду, миллиметрах в минуту, в секундах. Если единицы измерения скорости не заданы, то скорость устанавливается в процентах.

Параметр rotational speed – скорость вращения.

Параметр ALWAYS – всегда. Если этот параметр введен, скорость остается действительной до выполнения следующей инструкции SPEED. Если параметр не введен, скорость эффективна только для следующей инструкции движения (команды).

Пример – SPEED 50 – устанавливает скорость следующего движения 50 %; SPEED 100 – устанавливает скорость следующего движения 100 %; SPEED 200 – устанавливает скорость следующего движения 100 % (скорость свыше 100 % понимается как 100 %); SPEED 20MM/S ALWAYS – скорость движения начала инструментальной системы координат установлена 20 mm/s до тех пор, пока она не изменится другой командой SPEED, когда мониторинговая скорость 100 %.

Программные инструкции (команды) управления фиксаторами (схватами).

Команды: OPEN, OPENI.

Форматы команд:

OPEN номер фиксатора;

OPENI номер фиксатора.

Назначение команд. Открывает фиксаторы робота (открывает хват). Команда выдает сигналы для контролирования клапанов пневматического схвата при открытии фиксатора.

Параметр номер фиксатора. Выбирается номер фиксатора (схвата). Если параметр пропущен, выбирается первый фиксатор.

С командой OPEN сигнал не выйдет до тех пор, пока не начнется следующее движение. С командой OPENI, если робот находится в движении, сигнал выводится после того, как движение закончено.

Пример – OPEN – сигнал открытия фиксатора послан для контроля

клапана фиксатора 1, когда робот начинает следующее движение. OPEN1 2 – сигнал открытия фиксатора послан для контроля клапана фиксатора 2, как только завершилось текущее движение.

Команды: **CLOSE, CLOSE1.**

Форматы команд:

CLOSE – номер фиксатора;

CLOSE1 – номер фиксатора.

Назначение команды. Закрывает фиксаторы робота (закрывает хват). Команда выдает сигналы для контролирования клапанов пневматического схвата при закрытии фиксатора.

С командой CLOSE сигнал не выйдет до тех пор, пока не начнется следующее движение. С командой CLOSE1, если робот находится в движении, сигнал выводится после того, как движение закончено.

Пример – CLOSE 3 – сигнал закрытия фиксатора послан для контроля клапана фиксатора 3, когда робот начинает следующее движение. CLOSE1 – сигнал закрытия фиксатора послан для контроля клапана фиксатора 1, как только завершилось текущее движение.

Мониторные команды.

Команда **EDIT** запускает программу.

Формат команды:

EDIT имя программы.

Пример – >EDIT prog

Команда **E** – выход из режима редактирования и возврат в мониторный режим (конец программы Exit).

Команда **POINT.**

Формат команды:

POINT имя переменной позиции = значение позиции.

Назначение команды. Присваивает переменной позиции (точке), имеющей имя и стоящей слева в равенстве, значение координат позиции (точки), имеющей имя и стоящей в правой части равенства.

Параметр имя переменной позиции. Определяет имя точки, которую необходимо задать в декартовых или угловых координатах.

Параметр значение позиции. Задает имя реально существующей позиции (точки). Если параметр не задан, на дисплее появляются координаты точки с запросом на изменение Change?. При этом можно изменить координату точки, поставив в соответствующие координаты необходимые значения. Выполнение команды и ее закрытие происходит путем нажатия клавиши **Enter**.

Пример – >POINT #park

| | | | | | |
|--------|--------|--------|-------|--------|--------|
| JT1 | JT2 | JT3 | JT4 | JT5 | JT6 |
| 10.000 | 15.000 | 20.000 | 0.000 | 30.000 | 90.000 |

Change? (if not, hit RETURN only) (Enter)

„,35

| | | | | | |
|--------|--------|--------|--------|--------|--------|
| JT1 | JT2 | JT3 | JT4 | JT5 | JT6 |
| 10.000 | 15.000 | 20.000 | 35.000 | 30.000 | 90.000 |

Change? (if not, hit RETURN only) (Enter)

>POINT pick1 = pick (Enter) – присваивает точке pick1 значение точки pick в декартовых координатах.

>POINT pos0 = #pos0 (Enter) – преобразует точку #pos0 в угловых координатах в декартовую и присваивает точке pos0 значения точки #pos0.

Команда HERE.

Формат команды:

HERE имя переменной позиции.

Назначение команды. Задает имя переменной позиции для текущего положения, запоминает координаты точки в пространстве. Координаты могут быть декартовыми, угловыми.

Параметр имя переменной позиции присваивает заданной точке имя.

Имя должно начинаться с буквы или префикса #.

Пример – HERE #pick – запоминает точку в угловых координатах под именем pick. HERE place – запоминает точку в декартовых координатах под именем place.

Создание программы на языке AS.

Робот выполнит следующую задачу: поднимет деталь, подаваемую движущимся транспортером, и переместит ее в коробку (рисунок 4.1).

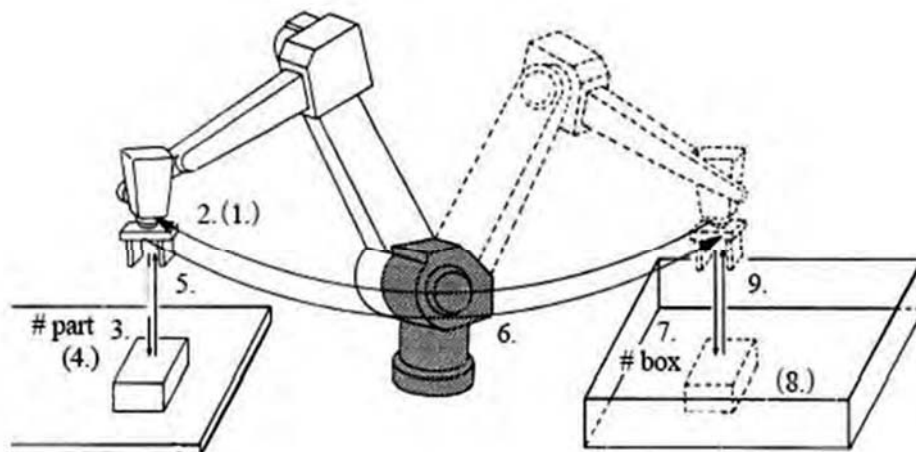


Рисунок 4.1 – Схема последовательности действий робота

Последовательность действий, требуемых для выполнения задачи.

- 1 Проверить, является ли схват открытым.
- 2 Переместиться в положение на 50 мм выше детали на транспортере.
- 3 Переместиться прямо вниз к детали в точку 3 (точка описывается переменной – #part).
- 4 Взять деталь и закрыть схват.

- 5 Переместиться прямо вверх на 150 мм транспортера.
- 6 Переместиться в позицию на 200 мм выше коробки.
- 7 Переместить деталь вниз в коробку в точку 7 (точка описывается переменной – #box).
- 8 Открыть схват и освободить деталь.
- 9 Переместиться снова в позицию на 180 мм выше коробки.

Код программы:

```
>EDIT demo
.PROGRAM demo () 1?OPENI
2?JAPPRO #part, 50
3?LMOVE #part 4?CLOSEI
5?LDEPART 150
6?JAPPRO #box, 200
7?LMOVE #box
8?OPENI
9?LDEPART 180
10?E
```

E – команда завершает режим редактирования. Когда программа выполняется, AS система выполняет шаги в порядке от шага 1 до шага 9.

Для обслуживания роботом технологического оборудования РТК в таблице 4.1 приведены сигналы управления и их значения для портов контроллера.

Таблица 4.1 – Сигналы управления для портов выхода контроллера робота (внешние сигналы)

| Назначение сигнала | Логическое имя порта выхода | Возможное значение сигнала |
|-------------------------------------|-----------------------------|-----------------------------|
| Дверь кабины рабочей зоны станка | Do3 | 0 – закрыть, 1 – открыть |
| Токарный патрон | Do4 | 0 – отжать, 1 – зажать |
| Выполнение программы точения детали | Do7 | 1 – выполнить |

Задания к практическому занятию

1 Изучить:

- основы программирования робота;
- системы координат, схему осей и рабочую зону робота;
- программные инструкции (команды) движения робота;
- программные инструкции (команды) двоичных сигналов;
- программные инструкции (команды) управления скоростью перемещения звеньев робота;
- программные инструкции (команды) управления схватами;
- мониторинговые команды;
- пример создания программы на языке AS;
- пример программы для обслуживания роботом технологического оборудования РТК.

2 Разработать программу для выполнения роботом транспортирования заготовок согласно варианту. Задания выдаются преподавателем.

Контрольные вопросы к практическому занятию № 7

- 1 Какой язык программирования используется для разработки программ для робота?
- 2 Назовите основные мониторные команды.
- 3 Назовите основные программные инструкции.
- 4 Приведите пример записи угловой координаты.
- 5 Приведите пример записи декартовой координаты.

5 Практическое занятие № 8. Изучение функций ввода команд программ с помощью редактора KRterm. Отладка и выполнение программ роботом

Типы программ в AS-системе.

Программа – это последовательность программных инструкций (команд), описывающих движение робота, функционирование внешних, внутренних сигналов, различные вычисления. Программное имя содержит не более 15 символов, начиная с алфавитной буквы, может продолжаться буквами и цифрами. Можно создать столько программ, сколько позволяет память контроллера. Программы создаются в ручном режиме (системный режим редактирования, блочное программирование) и с помощью программного обеспечения, в частности KRterm.

Последовательность ввода программ.

Редактор KRterm используется для создания и редактирования программ. Чтобы создать программу с именем demo нужно ввести EDIT demo и нажать клавишу Enter. Появится экран, отображающий следующее.

```
>EDIT demo PROGRAM demo 1?
```

Теперь система ожидает ввода первого шага. Первый шаг может быть таким: вводим **OPENI** после **1?** и нажимаем Enter.

```
> EDIT demo PROGRAM demo 1?OPENI [Enter]
2?
```

Затем можем ввести команду **JAPPRO #part, 50** для второго шага и нажимаем Enter.

```
>EDIT demo PROGRAM demo 1?OPENI
2?JAPPRO #part, 50 [Enter] 3?
```

Вводим остальную часть программы тем же самым способом. Исправлять ошибки при вводе шагов можно, нажимая клавишу Backspace только перед нажатием клавиши Enter. Если клавиша Enter нажата в конце ошибочного шага, появляется сообщение об ошибках и этот шаг бракуется. В этом случае нужно ввести шаг снова.

Команды редактирования.

Для редактирования программ в редакторе KRterm используются следующие команды (таблица 5.1).

Таблица 5.1 – Команды редактора

| Название команды | Описание команды |
|------------------|---|
| EDIT | Запускает программу редактора |
| C | Заканчивает редактирование текущей программы и заменяет ее другой программой (Change) |
| S | Выбирает программный шаг для отображения (Step) |
| P | Отображает заданное количество программных шагов (Print) |
| L | Выбирает предыдущий шаг (Last) |
| I | Вставляет новый шаг (Insert) |
| D | Удаляет программные шаги (Delete) |
| R | Заменяет символы (Replace) |
| O | Размещение курсора в текущем шаге (One line) |

Описание команд редактирования.

Команда **C**.

Формат команды:

C имя программы, номер шага

Назначение команды. Заменяет программу, находящуюся в редакторе, на другую.

Параметр имя программы. Определяет программу для редактирования.

Параметр номер шага. Выбирает номер шага, с которого начинается редактирование. Если номер шага не задан, редактирование начинается с первого шага.

Команда **S**.

Формат команды:

S номер шага

Назначение команды. Выбирает и отображает шаг программы, выбранный для редактирования.

Параметр номер шага. Если номер шага не задан, происходит переход на первый шаг программы. Если номер шага больше, чем шагов программы, происходит переход на шаг, следующий за последним шагом программы.

Команда **P**.

Формат команды:

P количество шагов

Назначение команды. Отображает число шагов программы, заданных в параметре, начиная с текущего шага. Последний шаг в списке готов к редактированию.

Параметр количество шагов. Задаёт количество шагов для просмотра.

Если количество шагов не задано, отображаются все шаги, следующие за текущим шагом.

Команда **L**.

Назначение команды. Отображает предыдущий шаг.

Команда I.

Назначение команды. Вставляет шаг перед текущим шагом. Заканчивается ввод команды нажатием клавиши Enter. Шаги после вставленной строки перенумеровываются.

Пример – CLOSEI команда вставлена между шагами 3 и 4.

```
1?OPENI
2?JAPPRO #PART, 500
3?LMOVE #PART
4?LDEPART 1000
5?S 4; отображает шаг 4 для вставки шага перед ним
4LDEPART 1000
4?I ; Type the I command (введите команду I)
4I CLOSEI; вставляем команду CLOSEI
5I [Enter]; нажимаем ввод для завершения вставки шагов
5LDEPART 1000; step 4 (шаг 4-ый стал пятым)
5?
```

Команда D.

Формат команды:

D количество шагов

Назначение команды. Удаляет заданное число шагов, следующих за текущим шагом.

Параметр – **количество шагов**. Задаёт число шагов для удаления, начиная с текущего шага. Если заданное количество шагов больше шагов программы, удаляются все шаги, следующие за текущим шагом. После удаления все оставшиеся шаги автоматически перенумеровываются и сдвигаются вверх.

Команда R.

Формат команды:

R символьная цепочка

Назначение команды. Заменяет ранее введенные символы в данном шаге на последовательность заданных символов.

Параметр **символьная цепочка**. Задаёт новые символы для замены ранее введенных символов.

Последовательность действий для использования команды **R** следующая.

1 Используя клавишу пробел **Spacebar**, установить курсор под первым символом, который необходимо изменить.

2 Нажать клавишу **R** и затем **Spacebar**.

3 Ввести новый символ замены.

4 Нажать клавишу **Enter**.

Пример – Изменить скорость движения с 20 % на 35 %, используя команду **R**.

```
1 SPEED 20, ALWAYS
1? R 35 [Enter]
1 SPEED 35, ALWAYS
1?
```

Команда **O**.

Назначение команды. Устанавливает курсор за последним символом выбранного шага.

Пример – Изменить имя точки aa1 на cc1, используя команду **O**.

3 JMOVE aa1

3?O

3 JMOVE aa1

3 JMOVE aa1 [BackSpace]; удаляем aa1 3 JMOVE cc1 [Enter]; вводим cc1

3?

Запуск редактора KRterm (терминала) для ввода команд.

Запись программы в виде команд (программных инструкций и мониторных команд) выполняется в следующей последовательности.

1 Открыть программу (терминал) KRterm.

2 Подключиться к контроллеру. В окне (рисунок 5.1) выбрать Com/имя (первая строчка).

3 Ввести логин as.

4 Ввести команду EDIT имя программы.

5 Ввести команды – программные инструкции.

6 Ввести мониторные команды.

7 Выполнить введенную программу.

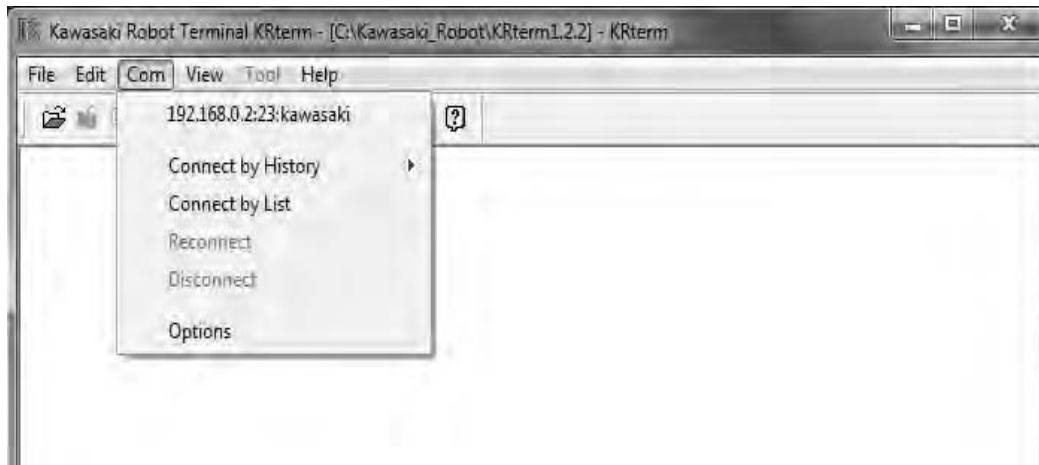


Рисунок 5.1 – Окно терминала редактора KRterm

Выполнение управляющих программ робота.

После ввода команд программы в редакторе KRterm программа автоматически записывается в память контроллера. Редактор с кодом введенной программы можно не закрывать при ее выполнении.

Для выполнения программы нужно установить переключатель **TEACH/REPEAT** на контроллере в положение **REPEAT**. Переключатель **TEACH LOCK** на пульте ручного управления должен находиться в положении **OFF**. Затем нужно включить силовое питание привода на пульте ручного управления нажатием клавиш **A + Motor ON** и **A + RUN**.

Мониторная скорость должна быть ниже 30 %. Вначале нужно установить скорость 10 %.

>SPEED 10 [Enter]

Скорость можно установить и в ручном режиме нажимая клавиши **A + SPEED**. Если робот движется корректно со скоростью 10 %, то ее можно увеличить. Если движения робота выполняются некорректно, то нужно нажать клавиши пульта ручного управления **A + HOLD**. Робот замедлится и остановится. В случае аварийной ситуации нужно нажать кнопку аварийного останова **EMERGENCY STOP** (красного цвета) на панели контроллера или на пульте ручного управления. Тормозная система робота сработает, и робот остановится немедленно.

Для того чтобы начать выполнение программы, можно использовать команду EXECUTE.

>EXECUTE demo [Enter]

Робот должен выполнить программу.

>SPEED 30 [Enter]

>EXECUTE demo [Enter]; *робот движется со скоростью 30 %.*

После того как программа EXECUTE была выполнена не менее одного раза, можно использовать клавиши **A + Cycle Start** пульта ручного управления.

Чтобы выполнить программу не один раз, нужно ввести номер повторений после имени программы.

EXECUTE demo,5 [Enter]; *выполняет программу 5 раз*

EXECUTE demo,-1 [Enter]; *выполняет программу непрерывно*

Выполнение программы через мониторную команду STEP или клавишу пульта управления **CHECK** дает возможность проверить движение и содержимое программы при выполнении шаг за шагом.

При записи команд в редакторе KRterm координаты точек можно вводить, используя мониторные команды POINT и HERE, или обучить точки в ручном режиме. При обучении точек нужно подвести схват робота к требуемой точке в пространстве в ручном режиме, потом ввести в редакторе KRterm:

>HERE имя точки [Enter]

Координаты точек появятся на экране.

Выполнение программы можно осуществлять и в ручном режиме. Для этого нужно выбрать программу, используя пульт ручного управления. Необходимо выбрать из пошагового меню функцию Directory (рисунок 5.2). После появится экран выбора программы (рисунок 5.3), и из списка следует выбрать программу.

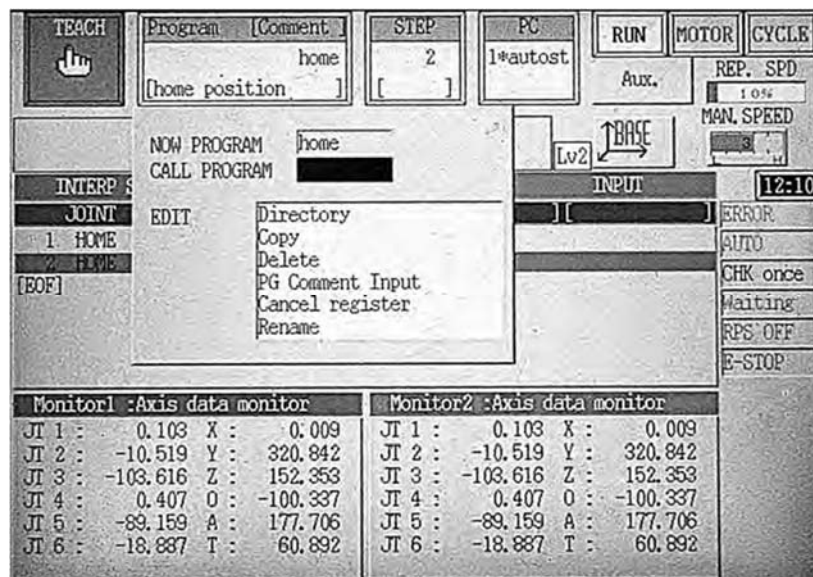


Рисунок 5.2 – Экран выбора функции Directory на пульте ручного управления

| Directory | | Program Name : 1 |
|-----------------|----------|------------------|
| PROGRAM NAME | STEP NUM | COMMENT |
| 1 | 0 | () |
| autostart.pc | 155 | () |
| demo | 31 | () |
| get_picture | 12 | () |
| home | 2 | home position) |
| kaw1 | 16 | home position) |
| kaw2 | 10 | home position) |
| kaw3 | 0 | () |
| 899 | 5 | () |
| primer1 | 20 | () |
| prog1 | 2 | () |
| prog2 | 6 | () |
| prog3 | 17 | () |
| tcpcomm_close | 17 | () |
| tcpcomm_connect | 50 | () |
| tcpcomm_end | 1 | () |
| tcpcomm_recv | 38 | () |
| tcpcomm_send | 41 | () |

Input Next Page

Рисунок 5.3 – Экран Directory выбора программы на пульте ручного управления

Затем нужно запустить программу, установив переключатель **TEACH/REPEAT** на контролере в положение **REPEAT**, переключатель **TEACH LOCK** на пульте ручного управления – в положение **OFF** и нажать клавиши **A + Motor ON**, **A + RUN** и **A + Cycle Start**.

Останов выполнения программ.

Существует несколько способов остановки выполнения программы во время работы робота. Ниже приведены три способа остановки от самого срочного до менее срочного.

1 Нажимаем кнопку аварийного останова (красного цвета) **EMERGENCY STOP** на операционной панели контроллера или на пульте ручного управления. Тормозная система сработает, и робот немедленно остановится. До тех пор, пока нет аварийной ситуации, нужно использовать способы 2 или 3.

2 Нажимаем клавиши пульта ручного управления **A + HOLD**. Робот замедлится и остановится.

3 Ввод команды **ABORT** останавливает программное выполнение после того, как робот завершит текущий шаг (инструкцию движения).

>ABORT [Enter]

Для останова выполнения программы также может быть использована команда **HOLD**.

>HOLD [Enter]

Возобновление выполнения управляющих программ робота.

В зависимости от того, каким способом робот был остановлен, существует несколько способов возобновления выполнения программ.

1 Если робот был остановлен при помощи кнопки аварийного останова (красного цвета) **EMERGENCY STOP**, то нужно разблокировать эту кнопку и нажать клавиши пульта ручного управления **MOTOR POWER** для того, чтобы включить силовое питание привода. Робот начинает движение после нажатия клавиш **A + Cycle Start**.

2 Если для останова робота были использованы клавиши пульта ручного управления **A + HOLD**, то нужно нажать клавиши **A + RUN** для продолжения выполнения программы.

3 Для того чтобы продолжить выполнение программы после команд **ABORT** или **HOLD** или когда выполнение программы было прервано из-за какой-то ошибки, нужно использовать команду **CONTINUE** (ошибка должна быть сброшена до возобновления выполнения программы).

>CONTINUE [Enter]

Команды управления программой и данными. Команды сохранения программ и данных.

Команды управления программой и данными и команды сохранения программ и данных являются мониторными и записываются в редакторе **KRterm** после знака «>» (таблица 5.2).

Таблица 5.2 – Команды управления программой и данными

| Название команды | Описание команды |
|------------------|---|
| 1 | 2 |
| LIST | Отображает все программные шаги и значения переменных |

Окончание таблицы 5.2

| 1 | 2 |
|----------|--|
| LIST/P | Отображает все программные шаги, выводит код программы |
| LIST/L | Отображает все координаты и их значения |
| DELETE | Удаляет программы и переменные из памяти робота |
| DELETE/P | Удаляет программы из памяти робота |
| DELETE/L | Удаляет координаты из памяти робота |
| COPY | Копирует программы |
| SAVE/P | Сохраняет программы |
| SAVE/L | Сохраняет координаты |
| LOAD | Загружает программы и данные в память робота |

Пример программы, записанной в редакторе KRterm

```

This is AS monitor terminal "AUX1"
>EDIT prog3
  .Program prog3()
  1?HOME
  2?SPEED 50
  3?LMOVE tochka4
  4?LMOVE tochka5
  5?LMOVE tochka6
  6?SPEED 100
  7?LDEPART -200
  8?SPEED 30
  9?JMOVE tochka7
  10?JAPPRO tochka6, -200
  11?SPEED 100
  12?LAPPRO tochka6, 70
  13?SPEED 20
  14?LMOVE tochka6
  15?DELAY 5
  16?SPEED 100
  17?E
>POINT tochka4
  X[mm]    Y[mm]    Z[mm]    O[deg]    A[deg]    T[deg]
  0.000    0.000    0.000    0.000    0.000    0.000
Change? (If not, Press RETURN only.)
Program held.No = 1
-220,450,130,-100,177,61
  X[mm]    Y[mm]    Z[mm]    O[deg]    A[deg]    T[deg]
  -220.000  450.000  130.000  -100.000  177.000  61.000
Change? (If not, Press RETURN only.)

>POINT tochka5
  X[mm]    Y[mm]    Z[mm]    O[deg]    A[deg]    T[deg]
  0.000    0.000    0.000    0.000    0.000    0.000
Change? (If not, Press RETURN only.)
200,300,200(D1561) [Power sequence board]AC primary power OFF.

PC program aborted.No = 1001
Connecting to Kawasaki E Controller

```

Задания к практическому занятию

1 Изучить:

- последовательность ввода программ;
- команды редактирования;
- последовательность выполнения управляющих программ робота;
- последовательность останова выполнения программ;
- последовательность возобновления выполнения управляющих программ робота;
- команды управления программой и данными;
- команды сохранения программ и данных.

2 Выполнить запись программы в виде команд (программных инструкций и мониторных команд) в соответствии с заданием, выданным преподавателем.

3 Отредактировать программу и выполнить введенную программу.

Контрольные вопросы к практическому занятию № 8

1 Какие существуют типы программ для управления роботом?

2 Для чего используется редактор Krterm?

3 Назовите последовательность ввода нескольких команд для разработки программы.

4 Назовите команды редактирования программы в редакторе KRterm.

5 Перечислите последовательность команд для запуска программы в редакторе Krterm.

6 Практическое занятие № 9. Изучение среды программирования K-ROSET робота Kawasaki

Описание среды программирования K-ROSET.

K-ROSET – это программное обеспечение Kawasaki, которое представляет из себя виртуальный симулятор с широким функционалом. Этот инструмент для оффлайн программирования позволяет добавлять 3D-модели роботов Kawasaki, периферийного оборудования, роботизированных инструментов и заготовок в виртуальную среду и создавать. Программное обеспечение имитирует процессы реального контроллера робота Kawasaki и позволяет пользователю взаимодействовать с 3D-моделью робота через подвесные экраны и расположение кнопок обучения. Интерфейс приложения изображен на рисунке 6.1, описание рисунка – в таблице 6.1.

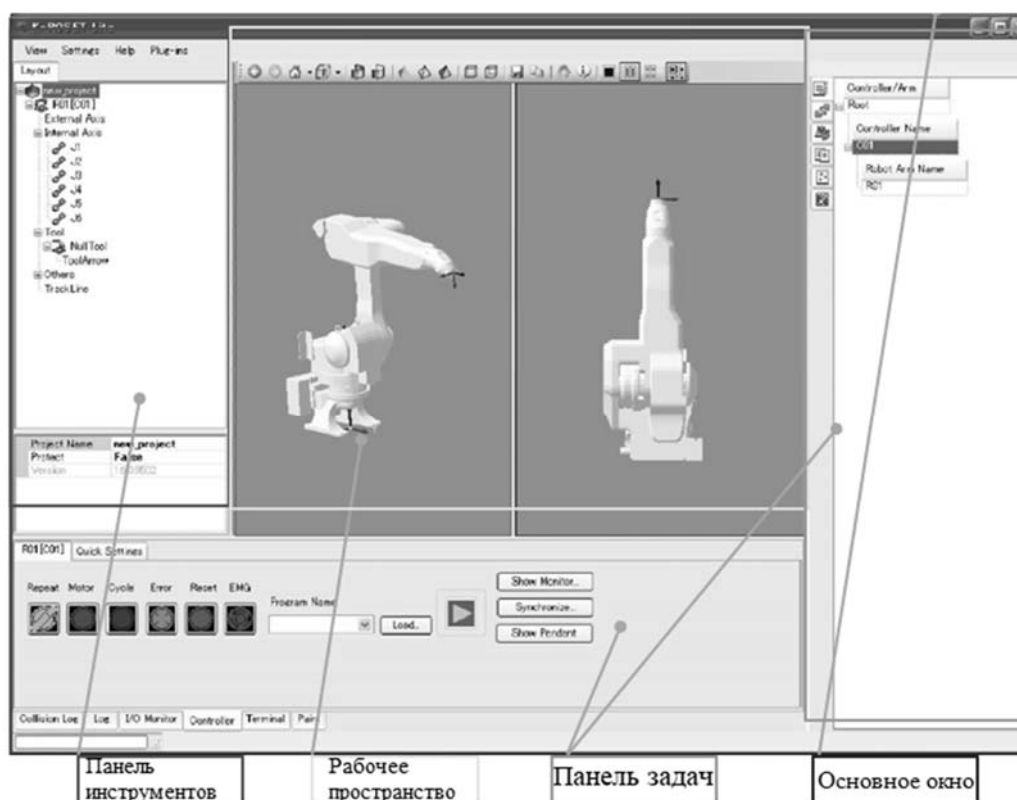


Рисунок 6.1 – Интерфейс приложения K-ROSET

Таблица 6.1 – Описание рисунка 6.1

| | |
|------------------------------|--|
| Основное окно (Main) | Главное окно интерфейса приложения |
| Панель инструментов (layout) | Основное меню, в котором происходит добавление и изменение модели |
| Рабочее пространство (View) | Поле, в котором отображается модель робота и его рабочего пространства |
| Панель задач (Task panel) | Меню, в котором отображается расширенная информация о программе, а также осуществляется программирование |

Создание проекта.

Для создания проекта на панели инструментов необходимо нажать ПКМ по иконке «New Project», после чего перейти на вкладку project – new. В появившемся окне необходимо вписать название проекта и, при необходимости, комментарии к нему (рисунок 6.2).

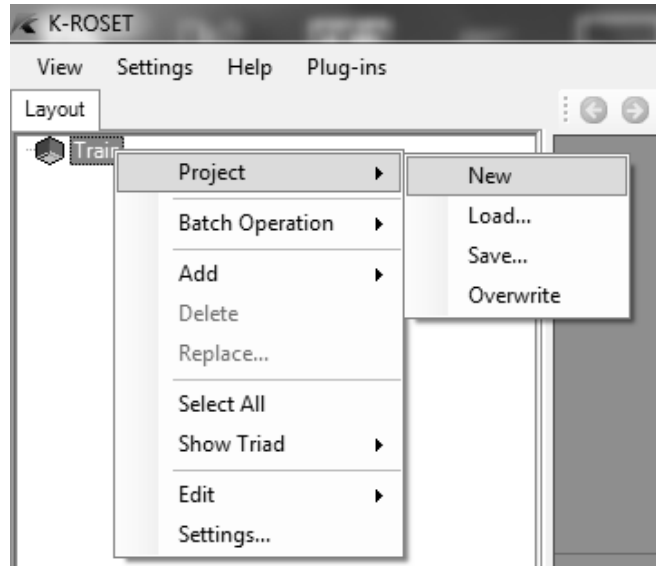


Рисунок 6.2 – Создание проекта

Для добавления робота в рабочее пространство необходимо нажать ПКМ по названию проекта и в появившемся меню перейти на вкладку Add -> Robot, после чего в появившемся окне выбрать необходимые параметры и нажать ОК (рисунок 6.3).

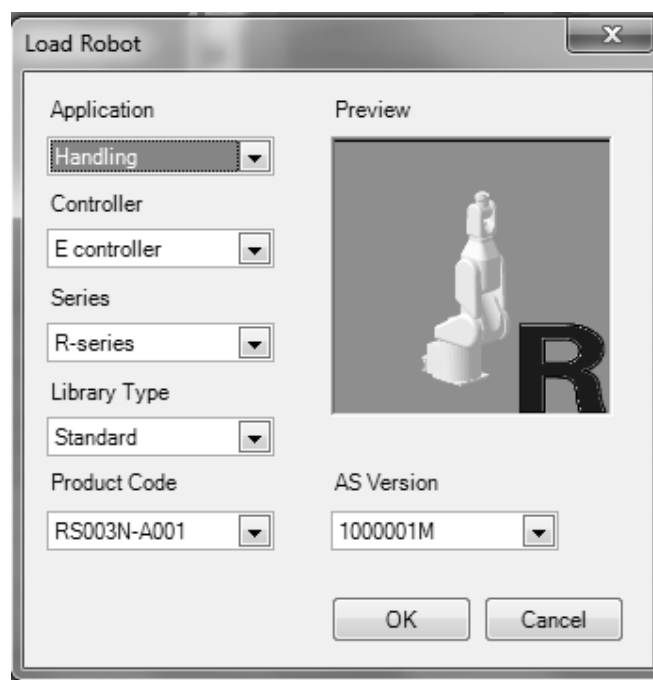


Рисунок 6.3 – Добавление робота

Аналогичным образом добавить модель рабочей области. На вкладке Add выбрать environment, затем work.

Далее необходимо разместить робота на рабочей области. Для этого требуется нажать по названию робота ПКМ перейти по вкладкам position > change.

В открывшемся меню можно выбрать шаг изменения координат как линейного, так и углового перемещения (рисунок 6.4).

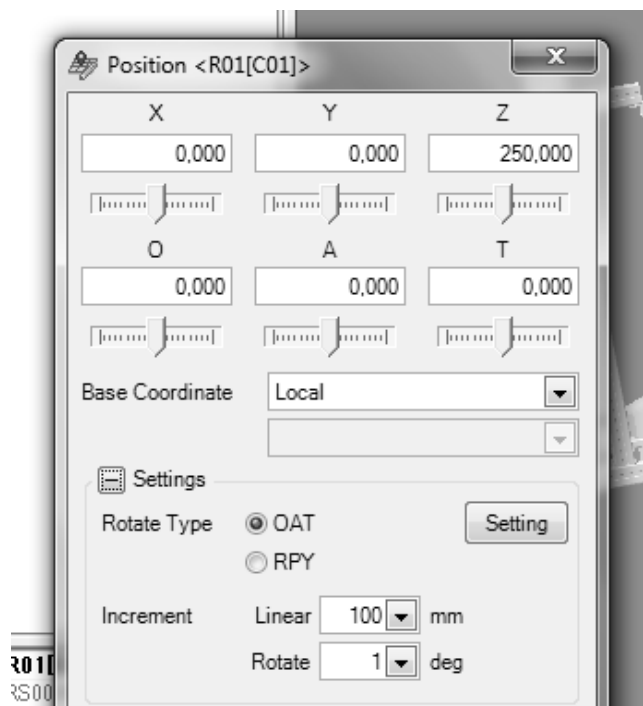


Рисунок 6.4 – Изменение положения робота

Создание программы.

Чтобы создать программу, необходимо в правом меню во вкладке program под надписью controller name кликнуть ПКМ по названию контроллера и перейти в раздел add – program (рисунок 6.5).

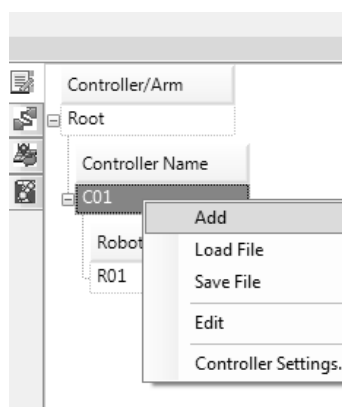


Рисунок 6.5 – Создание программы.

Далее нужно перейти на вкладку teach panel, в ней указаны координаты робота как в абсолютной, так и в угловой системе координат, которые можно

свободно менять, выбрав необходимый пункт меню и нажимая кнопки перемещения либо задавая конкретные координаты с клавиатуры.

Для создания точки задайте требуемые координаты и нажмите кнопку Add point.

Перемещая робота из точки в точку, создайте траекторию движения робота. Затем выберите точку, с которой хотите начать траекторию в левом меню.

Далее вернитесь в раздел program, нажмите ПКМ по названию программы, затем – add -> program step -> move и выберите необходимый тип перемещения, в данном случае JMOVE. Выбирая последующие точки повторить необходимое количество раз. В конце выбрать HOME (рисунок 6.6).

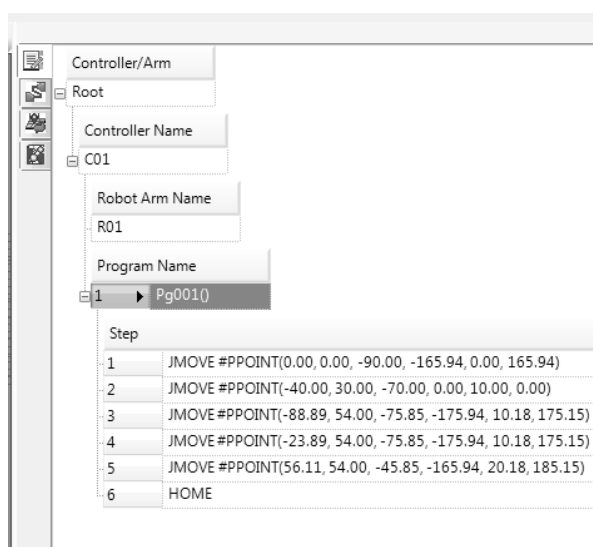


Рисунок 6.6 – Программа перемещения робота

В таблице 6.2 представлено описание основных команд, доступных в данном меню.

Таблица 6.2 – Описание команд

| Команда | Описание |
|--------------------------|---|
| 1 | 2 |
| Команды перемещения | |
| JMOVE | Движение с интерполяцией суставов |
| LMOVE | Движение с линейной интерполяцией |
| HMOVE | Движение с линейной интерполяцией и угловой интерполяцией сустава с рабочим органом |
| C1MOVE, C2MOVE | Движение с дуговой интерполяцией |
| Команды входов/выходов | |
| DIN WAIT | Ожидание входного сигнала |
| DOUT | Передача выходного сигнала |
| Команды задания скорости | |
| MOVING TIME | Задание времени перемещения (в секундах) |
| Команды ветвления | |

Окончание таблицы 6.2

| 1 | 2 |
|-------------------|---------------------------------------|
| Команды ветвления | |
| PGM SELECT | Переключение программы |
| GOTO | Переход по метке (без условий) |
| TRUE JUMP | Условный переход по метке (ИСТИНА) |
| FALSE JUMP | Условный переход по метке (ЛОЖЬ) |
| LABEL | Задание метки |
| DIN_JUMP | Сравнение входного сигнала с эталоном |

С более подробным списком команд и их назначением можно ознакомиться в Инструкции к программе K-ROSET (K-ROSET instruction manual) и языку программирования AS (AS language instruction manual).

После создания траектории необходимо проверить правильность выполнения программы. Для этого в программе требуется во вкладке execute выбрать пункт all steeps. После проверки правильности программы ее необходимо сохранить. Для этого нужно, перейдя во вкладку save files, выбрать пункт Program (рисунок 6.7).

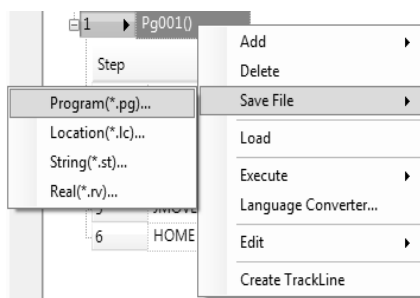


Рисунок 6.7 – Сохранение программы

Для загрузки уже заранее созданной программы необходимо в нижней части панели задач перейти на вкладку controller, затем перевести переключатель в положение repeat и нажать кнопку motor (рисунок 6.8).

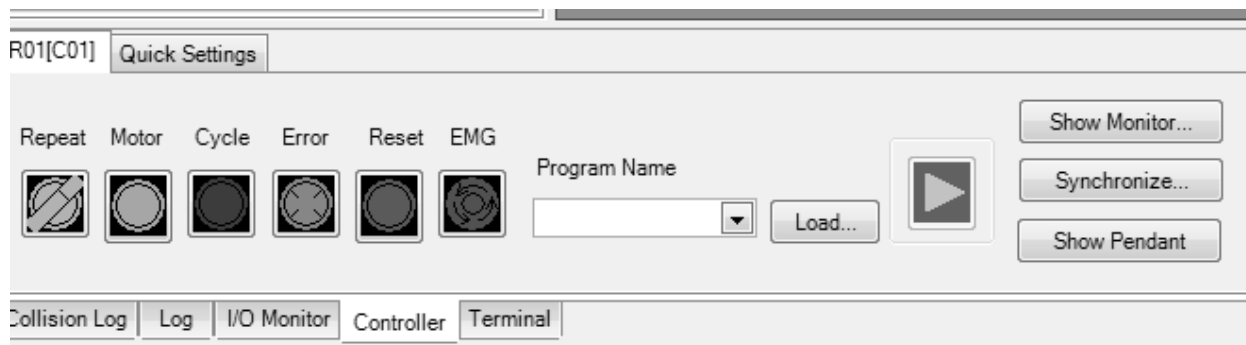


Рисунок 6.8 – Подготовка к загрузке готовой программы

После этого необходимо загрузить непосредственно саму программу. Для

этого требуется нажать кнопку Load и выбрать файл программы. Воспроизвести программу можно, нажав кнопку Пуск (рисунок 6.9).

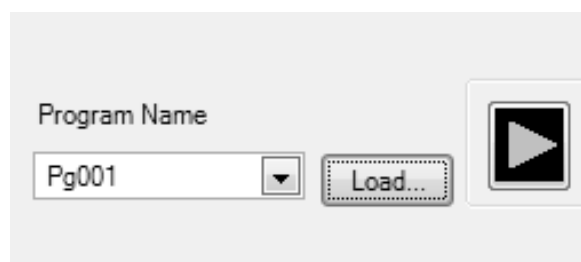


Рисунок 6.9 – Загрузка и запуск готовой программы

Задания к практическому занятию

1 Изучить принципы настройки виртуальной среды и программирования робота K-ROSET.

2 Выполнить выданное преподавателем индивидуальное задание по созданию программы работы робота.

Контрольные вопросы к практическому занятию № 9

1 Назовите основные элементы интерфейса программы K-ROSET.

2 Назовите основные типы команд.

3 Для чего служит команда HOME?

Список литературы

1 Основы робототехники: учебное пособие / Е. Е. Ступина [и др]. – Новосибирск: Сибпринт, 2019. – 160 с.

2 **Иванов, А. А.** Модернизация промышленных предприятий на базе современных систем автоматизации и управления: учебное пособие / А. А. Иванов. – Москва: ФОРУМ; ИНФРА-М, 2019. – 384 с.