# Написание программы для ручного управления роботом манипулятором

## **Режимы работы робота-манипулятора**

Промышленный робот-манипулятор может эксплуатироваться в нескольких режимах работы, которые переключаются с помощью физических переключателей и сенсорного экрана пульта управления. Ниже приведены доступные варианты выбора режимов:

* ручной режим с пониженной скоростью (T1) – для тестирования, отладки, программирования и обучения (максимально возможная скорость манипулятора вручную не превышает 250 мм/с);
* ручной режим с ограниченной максимальной скоростью (T2) – для тестирования, отладки и программирования (скорость ограничена более высоким значением, проводить обучение в этом режиме нецелесообразно);
* автоматический режим (AUT) – для промышленных роботов без системы управления верхнего уровня либо с выключенной системой. Робот выполняет действия в соответствии с запущенной программой, ручной режим управления невозможен (кроме аварийного останова);
* режим внешней автоматики (AUT EXT) – для промышленных роботов с активной системой управления верхнего уровня, реализованной, например, через ПЛК. Ручное управление роботом в этом случае невозможно (кроме аварийного останова).

**Системы координат**

В системе управления роботом заданы следующие декартовы системы координат (рис.1.1): ***WORLD*; *ROBROOT*; *BASE*; *TOOL*.**

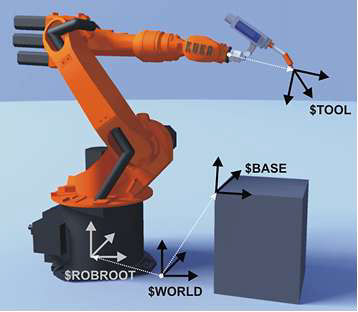


Рис.1.1. Общий вид систем координат

***WORLD***

Система координат *WORLD* представляет собой жестко заданную декартову систему координат. Она является исходной для систем координат *ROBROOT* и *BASE*. По умолчанию система координат *WORLD* расположена в основании робота.

ROBROOT

Декартова система координат *ROBROOT* всегда расположена в основании робота. Она описывает положение робота по отношению к системе координат *WORLD*. По умолчанию система координат *ROBROOT* совпадает с системой координат *WORLD*. Посредством *$ROBROOT* можно задать смещение робота относительно системы координат *WORLD*.

BASE

Система координат *BASE* представляет собой декартову систему координат, описывающую положение заготовки. Она соотнесена с системой координат *WORLD*.

По умолчанию система координат *BASE* совпадает с системой координат *WORLD*. Она перемещается пользователем в заготовку.

TOOL

Декартова система координат *TOOL* расположена в рабочей точке инструмента – TCP (Tool Center Point).

По умолчанию исходной точкой системы координат *TOOL* является центр фланца (в этом случае она называется системой координат *FLANGE*). Система координат *TOOL* перемещается пользователем в рабочую точку инструмента.

## **Калибровка**

Калибровка TOOL или числовой ввод (инструмент или заготовка на фланце)

Необходимые условия:

* + права пользователя: функциональная группа «Калибровка»;
  + режим работы *Т*1; остальные условия зависят от метода калибровки или значения *X*, *Y*, *Z*, *A*, *B* и *С* известны.

Порядок действий:

1. В главном меню выбрать пункт «*Ввод в эксплуатацию*> *Управление инструментами и базами*».
2. Выбрать вкладку «Инструмент, Заготовка».
3. Коснуться кнопки «Добавить». Открывается окно «Редактировать инструмент».
4. Присвоить номер и имя.
5. В поле справа от имени указать, что калибруется: «Инструмент» или «Заготовка».
6. В области «трансформация» задать значения для начала отсчета (*X*, *Y*, *Z*) и ориентации (*А*, *В*, *С*) системы координат *TOOL*:
   * ввести значения в полях или щелкнуть кнопку «Калибровка», выбрать метод калибровки и выполнить калибровку.
   * можно также ввести только значения для начала отсчета (*X*, *Y*, *Z*) или для ориентации (*А*, *В*, *С*) и определить остальные значения путем калибровки.
7. Сохранить.

В области «Последняя калибровка» теперь отображается информация об этой калибровке.

Калибровка BASE или числовой ввод (база или стационарный инструмент)

Необходимые условия:

* + права пользователя: функциональная группа «Калибровка»;
  + режим работы *Т*1;
  + остальные условия зависят от метода калибровки или известны значения *X*, *Y*, *Z*, *A*, *B* и *С*.

Порядок действий:

1. В главном меню выбрать пункт «*Ввод в эксплуатацию*> *Управление инструментами и базами*».
2. Выбрать вкладку «База, Стационарный инструмент».

3. Коснуться кнопки «Добавить». Открывается окно «Редактировать базу**»**.

1. Присвоить номер и имя.
2. В поле справа от имени указать, что калибруется: «База**»** или

«Стационарный инструмент».

1. В области «трансформация» задать значения для начала отсчета (*X*, *Y*, *Z*) и ориентации (*А*, *В*, *С*) системы координат *TOOL*:

ввести значения в полях или щелкнуть кнопку «Калибровка», выбрать метод калибровки и выполнить калибровку;

можно также ввести только значения для начала отсчета (*X*, *Y*, *Z*) или для ориентации (*А*, *В*, *С*) и определить остальные значения путем калибровки.

1. Сохранить.

В области «Последняя калибровка» теперь отображается информация об этой калибровке.

* 1. **Знакомство с языком KRL**

## **Структура программы KRL**

Структура программы *KRL* представлена на рис. 1.2.

Строка 1. Строка *DEF* показывает имя программы. Если программа представляет собой функцию, строка *DEF* начинается с «*DEFFCT*» и содержит также другие сведения. Строка *DEF* может быть отображена или затемнена.

Строка 2. Строка *INI* содержит инициализации для внутренних переменных и параметров;

Строка 4. Позиция *HOME*;

Строка 8. Движение *LIN*;

Строка 14. Движение *PTP*;

Строка 20. Позиция *HOME*;

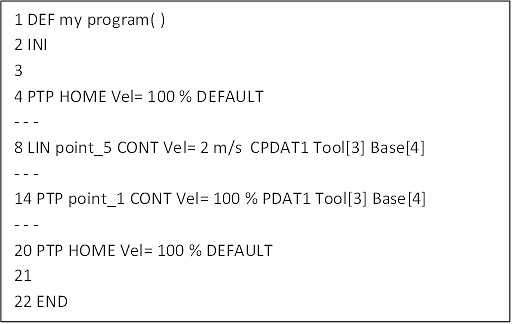


Рис. 1.2. Структура программы *KRL*

*Строка 22*. Строка *END* – это последняя строка в каждой программе. Если программа является функцией, строка *END* имеет вид «*ENDFCT*». **УДАЛЯТЬ СТРОКУ *END* ЗАПРЕЩЕНО!**

Первая команда движения в программе *KRL* должна содержать однозначное определение исходного положения. Это условие выполняется в позиции *HOME*, которая введена в систему управления роботом по умолчанию.

Если первой командой движения не является позиция *Default-HOME* или она была изменена, необходимо выполнить следующие команды:

* полностью команду *РТР* типа *POS* или *E*6*POS*;
* полностью команду *РТР* типа *AXISS* или *E*6*AXIS*.

«Полностью» означает, что должны быть указаны все компоненты целевой точки.

В программах, которые могут использоваться исключительно как подпрограммы, в качестве первой команды движения могут быть использованы также другие программы движения.

Позиция HOME

Позиция *HOME* действительна для всех программ. Она, как правило, используется в качестве первой и последней позиции в программе, так как она однозначно определена и некритична.

## **Редактирование программ**

Программирование движения РТР

Чтобы запрограммировать движение *PTP*, необходимо соблюдать следующие условия:

* права пользователя: функциональная группа «Старый внутри- строчный формуляр зоны перемещения»;
* программа выбрана;
* режим работы *T*1.

Порядок действий для программирования движения *PTP*:

1. Переместить точку *TCP* в положение, которое должно быть запрограммировано в качестве целевой точки.
2. Установить курсор в строку, после которой должна быть вставлена команда перемещения.
3. Выбрать последовательность меню «*Команды*> *Движение*> *PTP*»
4. Во встроенном формуляре установить параметры.
5. Сохранить команду, нажав кнопку «Команда OK»**.**

Встроенный формуляр PTP

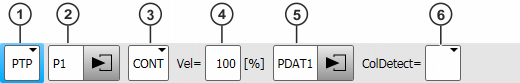


Рис. 1.3. Встроенный формуляр движения *РТР*

На рис. 1.3 обозначено:

* 1. – вид перемещения *РТР*;
  2. – имя целевой точки. Система автоматически задает имя. Имя может быть перезаписано. Для редактирования данных точки коснуться стрелки. Откроется соответствующее окно опции. С помощью стрелки можно редактировать также настройку «Глобальная точка»;
  3. – «*CONT*»: сглаживание целевой точки; [пусто]: точный подвод в целевую точку;

4 – скорость (1 … 100%);

1. – имя записи данных движения. Система автоматически задает имя. Имя может быть перезаписано. Для редактирования данных точки коснуться стрелки. Откроется соответствующее окно опции;
2. – распознавание столкновения для данного перемещения. «*OFF*»: функция распознавания столкновения отключена; «*CDSet*\_*Set*[*Nr*]»: функция распознавания столкновения включена. Для распознавания используются значения из набора данных *№*.

Программирование движения LIN

Программирование движения LIN осуществляется аналогично программированию движения PTP.

Программирование движения CIRC

Чтобы запрограммировать движение *CIRC*, необходимо соблюдать следующие условия:

* права пользователя: функциональная группа «Старый внутри- строчный формуляр зоны перемещения»;
* программа выбрана;
* режим работы *T*1.

Порядок действий для программирования движения *CIRC*:Переместить *TCP* в позицию, которая должна быть запрограмми- рована в качестве вспомогательной точки.

1. Установить курсор в строку, после которой должна быть вставлена команда перемещения.
2. Выбрать последовательность меню «*Команды*> *Движение*> *CIRC*».
3. Во встроенном формуляре установить параметры.
4. Нажать «*Touchup HP*».
5. Переместить точку *TCP* в положение, которое должно быть запрограммировано в качестве целевой точки.
6. Сохранить команду, нажав кнопку «Команда OK».

Встроенный формуляр перемещения *CIRC* отличается от формуляра движений *LIN/PTP* наличием вспомогательной точки (позиция 2 на рис. 1.4).

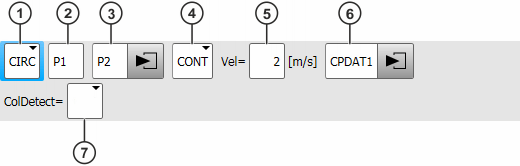


Рис.1.4. Встроенный формуляр перемещения *CIRC*

* 1. **Написание программы на языке KRL**

Алгоритм написания программы на языке KRL:

1. Создать программу на контроллере робота. Для этого необходимо перейти в директорию Program и далее нажать клавишу новый, выбираем шаблон модуль и жмем ОК. Далее задаем название программы.
2. Открываем созданную программу, нажатием кнопки выбрать. В исходной программе записаны две точки, это точки home – домашние позиции робота.
3. Следующим этапом вам необходимо переместить робота в 1 точку и запомнить ее, для этого необходимо перейти Команды>Перемещение>выбрать тип перемещения робота. В результате появится формуляр, который необходимо заполнить: выбрать инструмент, выбрать скорость перемещения робота. Сохраняем с помощью нажатия клавиши «Команда ОК».
4. Далее добавляете по аналогии все необходимые точки и команды управления захватом. Первым этапом необходимо добавить параметр остановки, который позволяет более безопасно захватить объект. Команды>Логика>Wait

Далее для закрытия захвата вида пальцы необходимо перейти в Команды>Логика>OUT>OUT. Для OUT1 устанавливаем значение True, добавляем OUT2 и устанавливаем ему значение False. Для открытия значения переменных OUT1 и OUT2 должны быть изменены на False и True соответственно.

Для управления захвата вида присоска необходимо перейти в Команды>Логика>OUT>OUT. Для OUT3 устанавливаем значение True в случае захвата и значение False в случае отпускания объекта.

1. После управления захватом прописываем команду Wait
2. Далее прописываем точки для перемещения объекта в его конечное положение
3. Возвращаемся в точку Home (в программе прописана)
   1. **Тестирование программы на двух манипуляторах**

Вам необходимо написать программы на двух манипуляторах: Kuka KR4 и Kuka KR6 и протестировать их в ручном режиме (T1) и на МИНИМАЛЬНОЙ скорости в автоматическом режиме (AUT) (при этом должны быть закрыты двери и соблюдены все требования техники безопасности)

**Общее для всех вариантов:** Написать, отладить и протестировать программу на языке KRL для выполнения конкретной технологической операции на манипуляторах Kuka KR4 и Kuka KR6.

**Общие требования:**

1. Программа должна иметь понятную структуру (строки DEF, INI, позиции движения, END).
2. Первой и последней командой движения должна быть позиция HOME или явно объявленная полная позиция.
3. Обязательно использование команд WAIT для пауз.
4. Тестирование должно быть проведено в режимах T1 и AUT (на минимальной скорости с соблюдением всех мер безопасности).

#### **Вариант 1: "Простая разгрузка"**

**Задача:** Написать программу, которая перемещает объект из точки А в точку Б по прямой траектории.

**Технологический процесс:**

1. Старт из позиции HOME.
2. Линейное движение (LIN) к точке над объектом в точке А.
3. Линейное движение вниз к точке захвата.
4. Закрытие пальцевого захвата (управление выходами OUT1 и OUT2).
5. Пауза WAIT 1 секунда.
6. Линейное движение вверх.
7. Линейное движение к точке над точкой Б.
8. Линейное движение вниз.
9. Открытие захвата.
10. Пауза WAIT 1 секунда.
11. Возврат в позицию HOME.

#### **Вариант 2: "Точное позиционирование"**

**Задача:** Используя движение PTP, точно установить деталь в гнездо.  
**Технологический процесс:**

1. Старт из HOME.
2. PTP движение к точке забора детали.
3. LIN движение для точного захвата.
4. Закрытие пальцевого захвата (управление выходами OUT1 и OUT2).
5. Пауза WAIT 2 секунды.
6. LIN движение вверх.
7. PTP движение к точке над гнездом.
8. LIN движение для установки детали.
9. Открытье пальцевого захвата (управление выходами OUT1 и OUT2).
10. Пауза WAIT 1 секунда.
11. Возврат в HOME через промежуточную точку PTP.

#### **Вариант 3: "Сборочная операция"**

**Задача:** Собрать конструкцию из двух деталей, находящихся в разных местах.  
**Технологический процесс:**

1. Старт из HOME.
2. Захватить первую деталь (пальцевый захват) из зоны 1.
3. Переместить ее и установить на сборочную площадку.
4. Открыть захват.
5. Переместиться к зоне 2.
6. Захватить вторую деталь (пальцевый захват).
7. Установить вторую деталь на первую.
8. Вернуться в HOME.

#### **Вариант 4: "Обход препятствия"**

**Задача:** Переместить объект по траектории с обходом условного препятствия.  
**Технологический процесс:**

1. Старт из HOME.
2. Захватить объект в точке Старт.
3. Поднять объект.
4. Выполнить движение PTP в точку "Высокая точка", чтобы обойти препятствие.
5. Выполнить движение LIN к точке над финишем.
6. Опустить объект.
7. Открыть захват.
8. Вернуться в HOME по прямой траектории (PTP).

#### **Вариант 5: "Циклическая операция"**

**Задача:** Написать программу, которая 3 раза выполняет операцию "взять-переместить".  
**Технологический процесс:**

1. Старт из HOME.
2. Используя цикл (или повторяя блок команд), выполнить 3 раза:
   * Подход к точке забора (LIN).
   * Захват (пальцевый захват).
   * Перемещение к точке сброса (PTP).
   * Открытие захвата.
3. После третьего цикла вернуться в HOME.

#### **Вариант 6: "** **Упаковка в тару "**

**Задача:** Написать программу для последовательного размещения трех одинаковых деталей в ячейках тары.

**Технологический процесс:**

1. Старт из позиции HOME.
2. Подход к точке забора детали (LIN).
3. Захват детали (пальцевый захват - OUT1=True, OUT2=False).
4. Подъем над точкой забора.
5. Перемещение к точке над первой ячейкой тары (PTP).
6. Опускание и размещение детали в ячейке 1 (LIN).
7. Открытие захвата.
8. Возврат к точке забора за следующей деталью.
9. Повторить шаги 2-8 для ячеек 2 и 3.
10. После упаковки третьей детали вернуться в позицию HOME.

### ***Вариант 7: "Контроль качества с отбраковкой"***

**Задача:** Реализовать программу, которая проверяет наличие детали и, в зависимости от результата, либо устанавливает ее, либо отправляет в брак.

**Технологический процесс:**

1. Старт из HOME.
2. Подход к конвейеру (зона контроля) (LIN).
3. Проверить сигнал датчика наличия детали (IN[1]).
4. ЕСЛИ IN[1] = TRUE (деталь есть), ТО:
   * Захватить деталь.
   * Пауза WAIT 1 сек.
   * Переместить к зоне качественных изделий.
   * Отключить вакуум.
5. ИНАЧЕ (детали нет или брак):
   * Подойти к зоне брака.
   * Имитация сброса бракованной детали (холостое движение LIN вниз-вверх).
6. Вернуться в позицию HOME.

#### **Вариант 8: "Сортировка по сигналу"**

**Задача:** Реализовать условное ветвление в программе на основе сигнала с цифрового входа.

**Технологический процесс:**

1. Старт из HOME.
2. Подойти к точке, где находится деталь.
3. Захватить деталь.
4. Проверить состояние цифрового входа IN[1].
5. ЕСЛИ IN[1] = TRUE, ТО перенести деталь в Зону А.
6. ИНАЧЕ перенести деталь в Зону Б.
7. Открыть захват.
8. Вернуться в HOME.

### ***Вариант 9: "Техническое обслуживание - имитация смазки"***

**Задача:** Написать программу, имитирующую процесс нанесения смазки в несколько точек условного механизма.

**Технологический процесс:**

1. Старт из HOME.
2. Активировать TOOL[10] (имитация шприца для смазки).
3. Подход к точке смазки №1 (PTP).
4. Касание точки (LIN), пауза WAIT 2 сек (имитация нанесения смазки), отвод.
5. Подход к точке смазки №2 (PTP).
6. Касание точки (LIN), пауза WAIT 2 сек, отвод.
7. Подход к точке смазки №3 (PTP).
8. Касание точки (LIN), пауза WAIT 2 сек, отвод.
9. Вернуться в HOME.

#### **Вариант 10: "Комплексная операция"**

**Задача:** Реализовать многоэтапный процесс, сочетающий все изученные элементы.  
**Технологический процесс:**

1. Старт из HOME.
2. Захватить заготовку (пальцевый захват) из зоны хранения.
3. Переместить ее к станцию обработки (PTP).
4. Установить заготовку (LIN).
5. Открыть захват.
6. Имитировать ожидание обработки (WAIT 3 сек).
7. Забрать обработанную деталь.
8. Переместить ее на контрольный стол (LIN).
9. Открыть захват.
10. Считать сигнал с датчика контроля (IN[2]).
11. ЕСЛИ IN[2] = TRUE (брак), ТО сбросить деталь в контейнер брака.
12. ИНАЧЕ перенести деталь на склад готовой продукции.
13. Вернуться в HOME.