

Московский авиационный институт
(национальный исследовательский университет)

**Автоматизация технологических процессов в
приборостроении с помощью промышленных роботов**

Учебное пособие

Москва, 2014

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

1. НАЧАЛО РАБОТЫ

- 1.1. Робот
- 1.2. Система управления роботом
- 1.3. Включение и выключение питания контроллера и робота
- 1.4. Режимы работы робота
- 1.5. Пульт обучения

2. УПРАВЛЕНИЕ РОБОТОМ В РУЧНОМ РЕЖИМЕ

- 2.1. Системы координат при ручном управлении
- 2.2. Система координат шарнира (JOINT)
- 2.3. Перемещение робота в ручном режиме
- 2.4. Некоторые коды ошибок

3. СОСТАВЛЕНИЕ ПРОГРАММЫ С ПОМОЩЬЮ ПУЛЬТА ОБУЧЕНИЯ

- 3.1. Регистрация программы с помощью пульта обучения
- 3.2. Команды регистра
- 3.3. Команды перемещения
 - 3.3.1. Формат движения
 - 3.3.2. Формат данных позиционирования
 - 3.3.3. Данные позиционирования
 - 3.3.4. Скорость перемещения шарнира
 - 3.3.5. Траектория позиционирования
 - 3.3.6. Ввод команды перемещения в режиме обучения
 - 3.3.7. Изменение данных позиционирования с помощью робота
 - 3.3.8. Изменение данных позиционирования с помощью экрана информации
 - 3.3.9. Изменение команды перемещения
 - 3.3.10. Позиционирование робота с помощью регистров положения
 - 3.3.11. Команда коррекции ускорения
- 3.4. Команды перехода
 - 3.4.1. Команда определения метки
 - 3.4.2. Команды безусловного перехода
 - 3.4.3. Команда конца программы и подпрограммы
 - 3.4.4. Использование аргументов в команде CALL
 - 3.4.5. Задание аргументов
 - 3.4.6. Команды условного перехода
 - 3.4.7. Команда условного выбора

4. КОМАНДЫ РЕДАКТИРОВАНИЯ ПРОГРАММЫ

- 4.1. Вставка пустых строк
- 4.2. Удаление команд
- 4.3. Копирование и вставка команд

5. ПРОГРАММИРОВАНИЕ ПЕРИФЕРИЙНЫХ УСТРОЙСТВ

- 5.1. Команды цифрового ввода-вывода робота
- 5.2. Команды цифрового ввода-вывода системы
- 5.3. Команды аналогового ввода-вывода системы

6. КОМАНДЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ

- 6.1. Запуск программы при помощи пульта обучения в пошаговом режиме
- 6.2. Запуск программы при помощи пульта обучения в непрерывном режиме
- 6.3. Запуск программы с панели оператора

1. НАЧАЛО РАБОТЫ

Робототехническая система FANUC (рис.) состоит из самого робота, блока управления, пульта обучения, соединительных кабелей и программного обеспечения ЧПУ.



Рис. Робототехническая система FANUC

1.1. РОБОТ

Робот - это механизм, состоящий из осей и рук, приводимых в движение серводвигателями. Точка, в которой руки соединяются, - это шарнир или ось.

Данная конфигурация робота имеет шесть осей: J1, J2, J3, J4, J5, J6.
(Рис.)

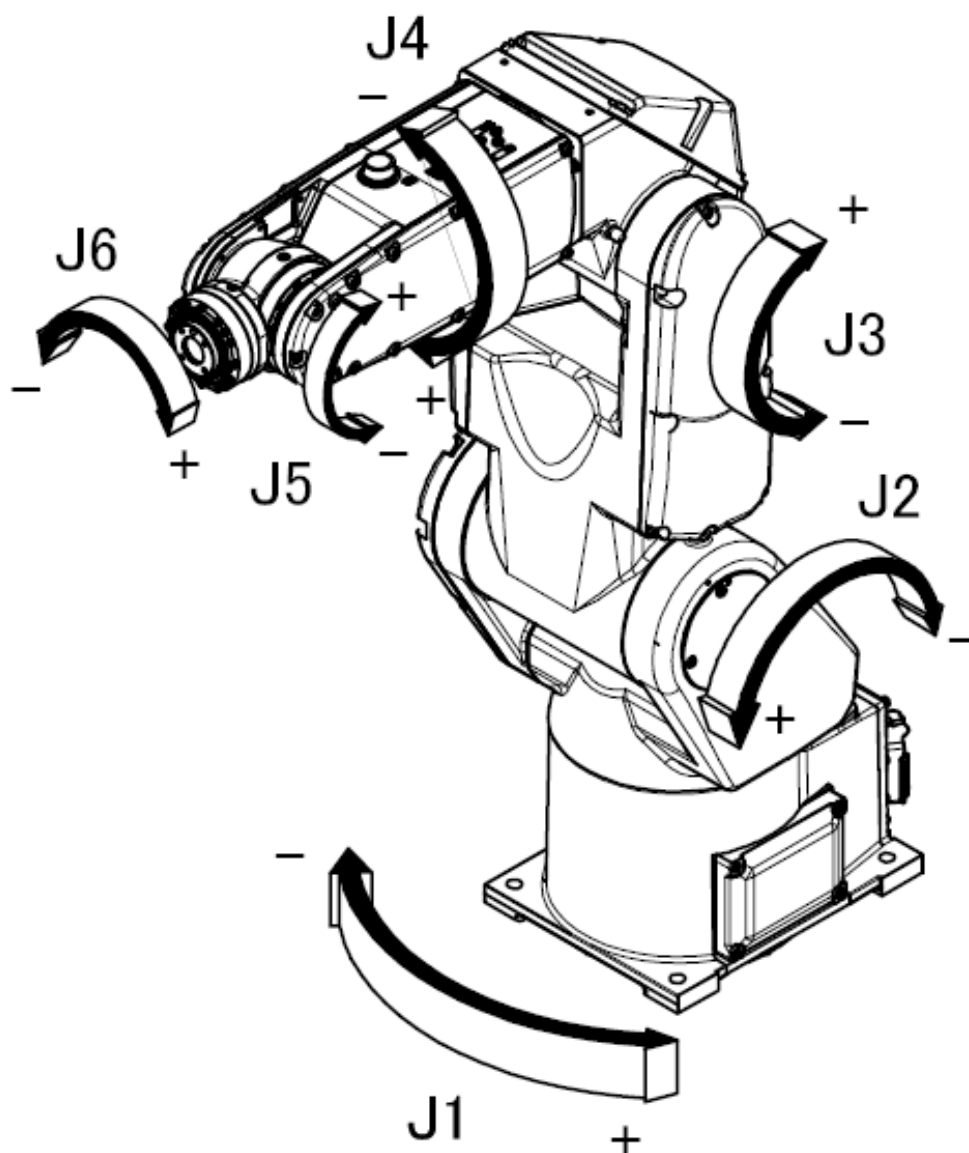


Рис. Кинематическая схема робота

1.2. СИСТЕМА УПРАВЛЕНИЯ РОБОТОМ

Система управления роботом (Рис.) предназначена для ручного и программно-логического управления роботом состоит из:

- контроллера робота ,
- пульта обучения.



Рис. .Общий вид контроллера и пульта обучения

Передняя панель контроллера называется панелью оператора (Рис.).



Рис. Панель оператора

На панели расположены:

1. Кнопка включения-выключения.
2. Трехпозиционный переключатель режимов работы робота.
3. Кнопка аварийной остановки
4. Коммуникационный порт USB.
5. Кнопка запуска программы в автоматическом режиме.
6. Замок дверцы контроллера.
7. Вентилятор.



1.3. ВКЛЮЧЕНИЕ И ВЫКЛЮЧЕНИЕ ПИТАНИЯ КОНТРОЛЛЕРА И РОБОТА

Включение питания контроллера и робота происходит при повороте переключателя на панели оператора по часовой стрелке в красный сектор (рис.). При включении контроллера питание подается и на сам робот.

Выключение питания контроллера и робота происходит при повороте переключателя на панели оператора против часовой стрелки в зеленый сектор. Робот при этом обесточивается. При дальнейшем повороте происходит разблокирование запорного механизма дверцы контроллера.



Рис. Выключатель питания контроллера

1.4.РЕЖИМЫ РАБОТЫ РОБОТА

Имеются три режима работы: AUTO (авто), T1 и T2. Они выбираются с помощью трехпозиционного переключателя.

Трехпозиционный переключатель - это переключатель с ключом, установленный на панели оператора (Рис.)

Этот переключатель используется для выбора оптимального режима работы робота согласно условиям отладки и эксплуатации.



Рис. Трехпозиционный переключатель режимов работы

T1 (<250 мм/с): тестовый режим 1

Режим T1 предназначен для:

- управления роботом в ручном режиме на малой скорости ,
- ввода координат положения с пульта обучения,
- проверки траектории движения робота на малой скорости,
- проверки управляющей программы.

Скорость как вершины инструмента, так и фланца ограничены и не превышают 250 мм/сек.

В режиме T1 невозможно проверить фактическую траекторию инструмента робота и время цикла, потому что рабочая скорость ограничена

Режим T1 можно зафиксировать, если извлечь ключ.

T2 (100%): тестовый режим 2

Режим T2 предназначен для окончательной проверки созданной программы путем прогона робота на скорости производства, потому что ,в принципе, ограничение скорости отсутствует. Однако, при ручном управлении и обучении скорость как вершины инструмента, так и фланца ограничены и не превышают 250 мм/сек.

Режим T2 можно зафиксировать, если извлечь ключ.

AUTO (авто): режим Auto (автоматический)

Режим AUTO предназначен для использования при промышленной эксплуатации робота.

Программа может выполняться только при запуске ее с *внешних устройств и панели оператора*. В режиме (АUTO), выполнение программы с пульта обучения невозможно.

Скорость работа при выполнении программы может выполняться на скоростях указанных в программе, в том числе и на максимальной. Однако, скорость как вершины инструмента, так и фланца ограничены и не превышают 250 мм/сек.

Использование данного режима для ручного управления и в режиме обучения невозможно.

Режим (АUTO) можно зафиксировать, если извлечь ключ.

1.5. ПУЛЬТ ОБУЧЕНИЯ

Пульт обучения (*Рис.*) обеспечивает интерфейс между прикладным программным обеспечением ЧПУ и оператором. Пульт обучения подключен кабелем к печатной плате в контроллере. Включает:

- жидкокристаллический дисплей (16 строк по 40 символов),
- набор клавиш,
- выключатель пульта обучения
- переключатель аварийной блокировки. Этот выключатель имеет три положения (отжатое, промежуточное, полностью нажатое). Перемещение робота может происходить только в промежуточном положении переключателя
- кнопку аварийной остановки. При нажатии кнопки аварийной остановки робот немедленно останавливается в любом режиме работы (Т1, Т2, АUTO).



Рис. . Пульт обучения. 1. Выключатель пульта обучения 2. Кнопка аварийной остановки 3. Переключатель аварийной блокировки

Передняя панель пульта обучения представлена на рис. .

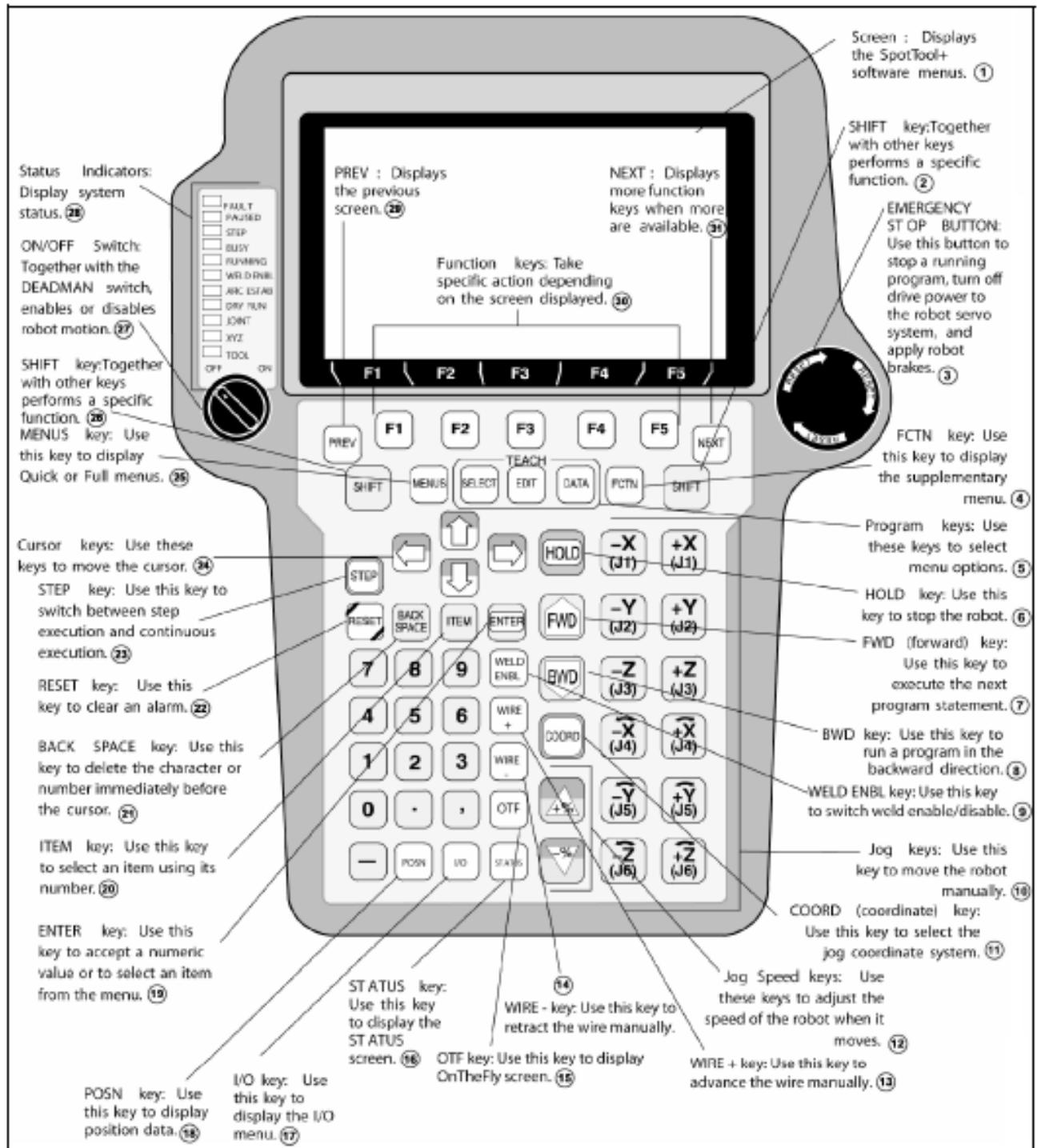


Рис. . Передняя панель пульта обучения

1. Экран: отображает программные меню SpotTool+.
2. Клавиша **SHIFT**: в сочетании с другими клавишами (отмеченными синим цветом) выполняет определенное действие.
3. КНОПКА АВАРИЙНОЙ ОСТАНОВКИ: для остановки выполняющейся программы, выключения питания управления сервосистемы робота и включения тормозов робота.
4. Клавиша **FCTN**: для отображения меню функций.
5. Программные клавиши: для выбора опций меню.
 - SELECT** (выбор) – для отображения экрана выбора программы.
 - EDIT** (редактировать) – экран редактирования программы.
 - DATA** (данные) – для отображения экрана данных программы.
6. Клавиша **HOLD**: используется для остановки робота.
7. Клавиша **FWD** (вперед): переход к следующему оператору программы.
8. Клавиша **BWD** (назад): переход к предыдущему оператору программы.
- 9.
10. Клавиши перемещения робота вручную.
11. Клавиша **COORD** (координата): клавиша выбора системы координат при ручном управлении.
12. Клавиши регулирования скорости при ручном управлении.
- 13.
- 14.
- 15.
16. Клавиша **STATUS**: отображает экран состояния.
17. Клавиша ввода-вывода: отображает меню ввода-вывода.
18. Клавиша **POSN**: отображает данные позиционирования.
19. Клавиша **ENTER**: используется для ввода числового значения или выбора пункта меню.
20. Клавиша **ITEM**: Используйте эту клавишу, чтобы выбрать позицию, используя ее номер. Перемещает курсор в строку, номер которой указан.
21. Клавиша **BACK SPACE**: удаляет слева от курсора.
22. Клавиша **RESET**: сбрасывает сигнал об ошибке.
23. Клавиша **STEP**: переключает пошаговое и непрерывное выполнение программы.
24. Клавиши курсора используются для перемещения курсора.
25. Клавиша **MENUS**: отображает экранное меню. Quick (быстрое) или Full (полное).
26. Клавиша **SHIFT**: в сочетании с другими клавишами, выделенные синим цветом. Используется для ручного управления роботом, обучения данным позиционирования и запуска программы. Правая и левая клавиши Shift работают одинаково..
27. Выключатель пульта управления.
- 28.

29. Клавиша **PREV**: восстанавливает предыдущее состояние экрана.
30. Кавиши **F1, F2, F3, F4, F5**:функциональные клавиши: для выбора меню функций перечисленных в последней строке экрана.
31. **NEXT**: клавиша следующей страницы – для переключения меню функциональных клавиш на следующую страницу , если таковая имеется.

2. УПРАВЛЕНИЕ РОБОТОМ В РУЧНОМ РЕЖИМЕ

В этом режиме ввод команд управления производится с пульта обучения вручную. При выполнении команд робот перемещается в целевое положение, которое впоследствии можно зарегистрировать.

Ручная подача зависит от следующих двух факторов:

--при ручном управлении задаются скорость перемещения робота и система координат для каждой степени подвижности.

--для задания скорости подачи необходимо нажать кнопку ручной коррекции (рис.).

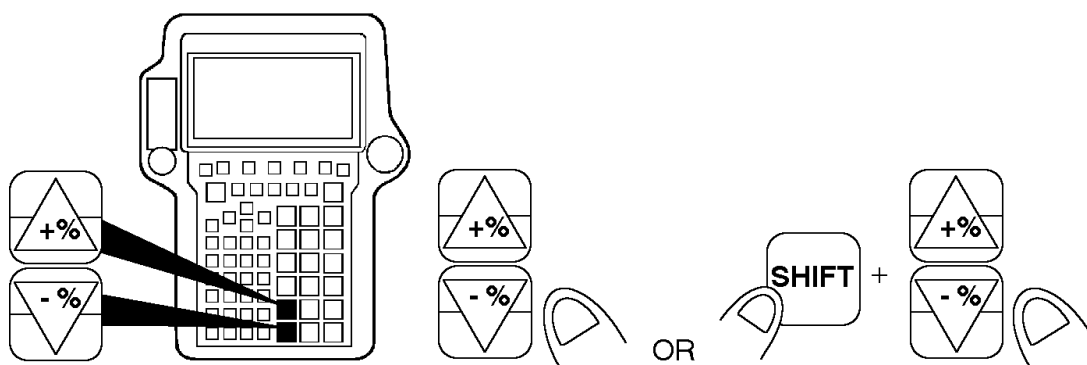


Рис. . Задание скорости подачи

При каждом нажатии клавиши ручной коррекции , скорость подачи последовательно изменяется в порядке:VFINE, FINE, от 1% до 5% с шагом 1% и от 50% до 100% с шагом 5%.

При каждом нажатии клавиши ручной коррекции при нажатой клавише SHIFT, скорость подачи последовательно изменяется в порядке: FINE, VFINE, 5%, 50% и 100%

Скорости перемещения, **представленные в процентах** (%) от максимально возможной в данном режиме и отображаются в верхнем правом углу экрана пульта обучения (рис.). Например, скорость подачи 100% означает, что робот перемещается с максимальной скоростью.

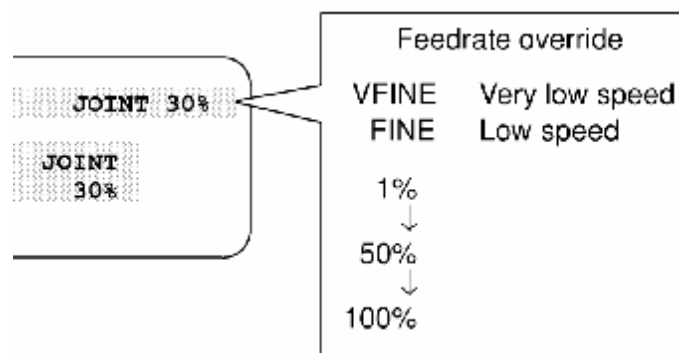


Рис. Вид экрана для задания скорости подачи

При скорости подачи **FINE** (тонкая, малая скорость перемещения) каждая ось поворачивается на 0,001 градуса при каждом нажатии кнопки перемещения.

Скорость подачи VFINE (самая тонкая, минимальная скорость перемещения) в десять раз меньше, чем при скорости **FINE** (тонкая).

Скорости подачи **FINE** и **VFINE** используются для более точного позиционирования рабочего органа

2.1. СИСТЕМЫ КООРДИНАТ ПРИ РУЧНОМ УПРАВЛЕНИИ

- Системы координат ручном управлении подразделяются на четыре типа:
- система координат шарнира (**JOINT**) (шарнир),
 - система декартовых координат (**XYZ**),
 - система координат инструмента (**TOOL**),
 - система координат пользователя (**USER**).

В данном пособии рассмотрена система координат шарнира (**JOINT**) (шарнир), как наиболее универсальная.

(Об остальных системах координат я напишу позже)

2.2. СИСТЕМА КООРДИНАТ ШАРНИРА (JOINT)

При использовании данной системы координат шарниры работа перемещаются независимо друг от друга (Рис.).

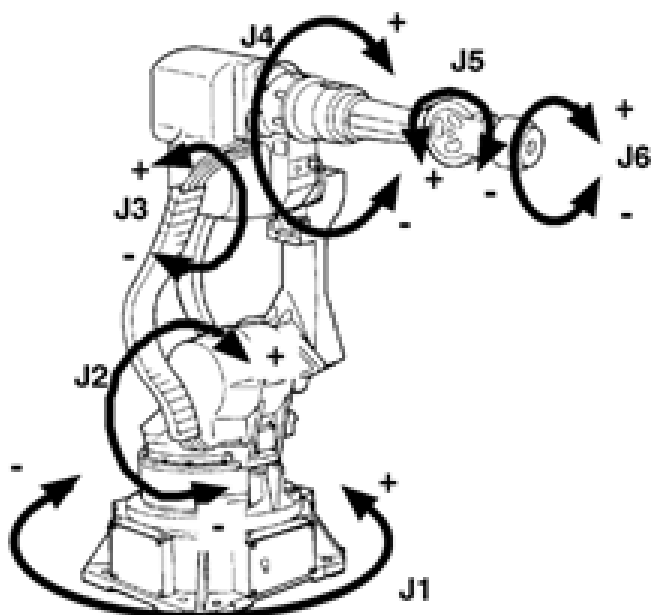


Рис. Система координат шарнира (JOINT)

Выбор системы координат JOINT

Текущая система координат ручной подачи отображается в верхнем правом углу экрана пульта обучения.

Для выбора системы координат нажмите кнопку COORD. При этом отображается всплывающее меню в верхнем правом углу экрана. (Рис.)

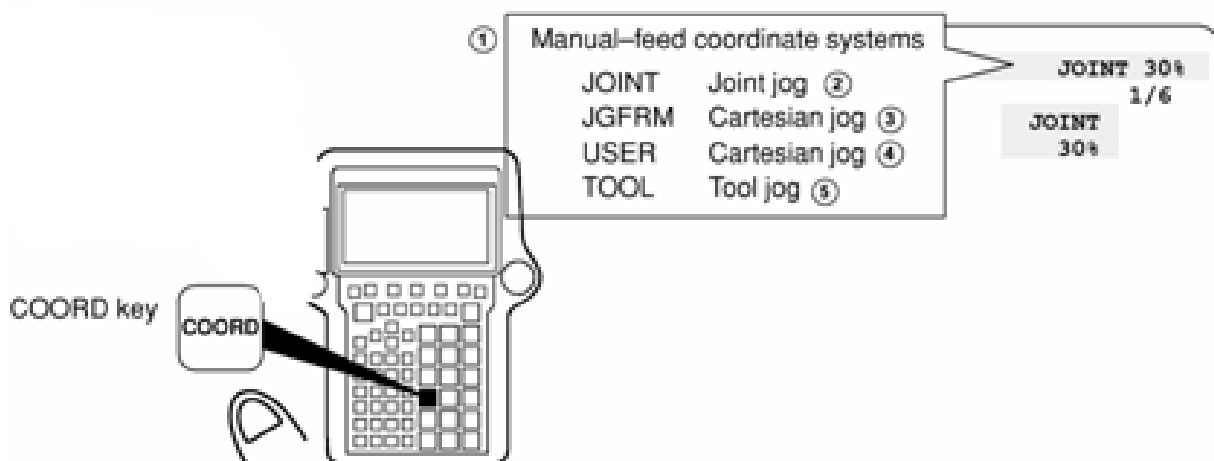


Рис. Экран выбора системы координат

При каждом нажатии клавиши COORD на пульте обучения выполняется смена системы координат ручной подачи в следующей последовательности JOINT -> JGFRM -> TOOL -> USER -> JOINT (выделяется цветом).

В нашем случае необходимо выбрать систему координат JOINT и нажать клавишу ENTER.

2.3. ПЕРЕМЕЩЕНИЕ РОБОТА В РУЧНОМ РЕЖИМЕ

Перемещение робота (в том числе и в ручном режиме) может происходить только при отсутствии ошибок. При наличии ошибок на пульте обучения горит красный индикатор и высвечивается номер ошибки

Для управления роботом в ручном режиме используются кнопки перемещения робота вручную при обязательном нажатии и удержании кнопки **SHIFT** (Рис.). При этом пульт управления должен быть включен, а переключатель аварийной блокировки сзади пульта обучения нажат (находиться в среднем положении). Какой из переключателей аварийной блокировки (правый или левый) будет нажат, не имеет значения.

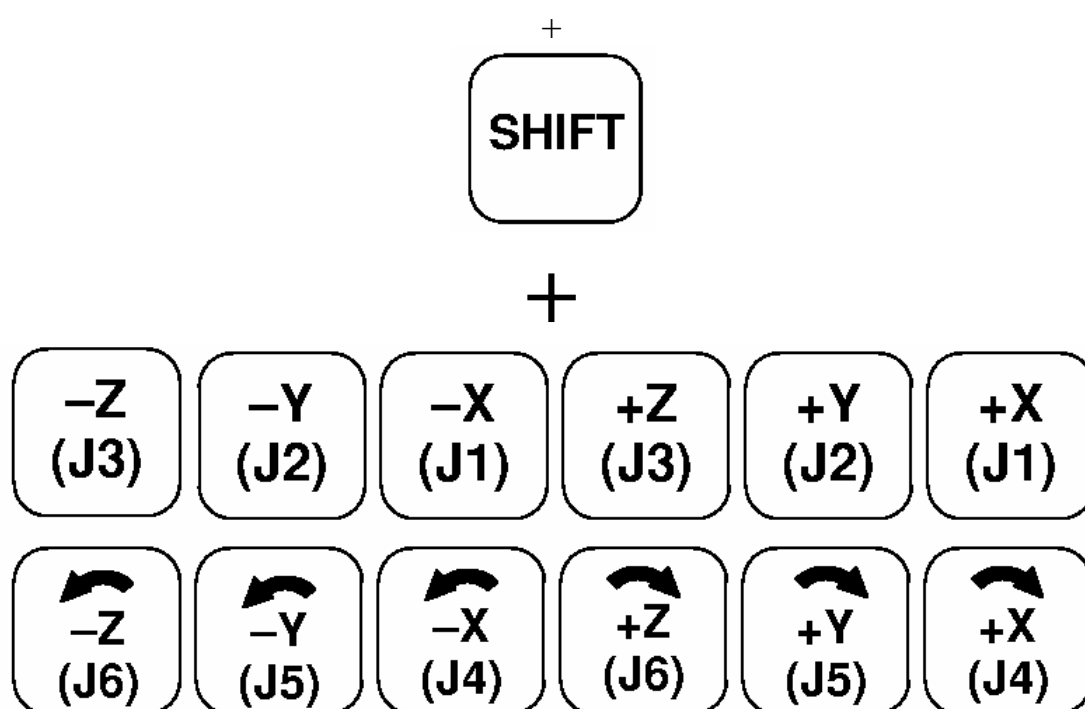


Рис. Клавиши перемещения при ручном управлении

Перемещение по осям (+ J1,- J1), (+ J2,- J2), (+J3, -J3), (+ J4,- J4), (+ J5, -J5),(+ J6,- J6) при ручном управлении может происходить только при отсутствии сигналов ошибки FAULT. Коды ошибок отображаются на экране пульта обучения.

2.4. НЕКОТОРЫЕ КОДЫ ОШИБОК

SRVO-001 Operator panel E-stop (Нажата аварийная кнопка на панели управления оператора)

SRVO-002 Teach pendant E-stop (Нажата аварийная кнопка на пульте обучения).

SRVO-003 DEADMAN switch released (Пульт обучения включен, но переключатель аварийной блокировки на тыльной стороне пульта не нажат или сильно нажат)

SRVO-004 Fence open (Открыто защитное ограждение)

SRVO-005 Robot overtravel (Робот вышел за пределы установленного диапазона движений)

1) На экране выключения выхода за пределы выберите [System OT release] для отключения сигнала выхода за пределы установленного ограничения каждой оси робота.

2) Удерживая нажатой клавишу **SHIFT**, для сброса сообщения о неисправности, нажмите на кнопку сброса.

3) Продолжая удерживать нажатой клавишу **SHIFT**, в шаговом режиме приведите оси в пределы заданного диапазона движений.

SRVO-006 Hand broken (Поломка кисти)

SRVO-105 Door open or E-stop (Открытая дверца шкафа контроллера)

После устранения ошибок необходимо при нажатом переключателе аварийной блокировки сзади пульта обучения нажать кнопку **RESET**. Индикация об ошибках на пульте обучения должна погаснуть.

Для отображения координат на экране пульта обучения каждой оси необходимо нажать кнопку **POSN (18)**.

3. СОСТАВЛЕНИЕ ПРОГРАММЫ С ПОМОЩЬЮ ПУЛЬТА ОБУЧЕНИЯ

Для управления роботом в отладочном, пошаговом и автоматическом режимах необходимо составить программу.

При этом ее предварительно необходимо зарегистрировать.

В данном пособии будут рассмотрены следующие команды управления:

- команды регистра,
- команды перемещения,
- команды ожидания,
- команды перехода,
- команды редактирования программы,
- команды программирования периферийных устройств,
- команды выполнения программы.

3.1. РЕГИСТРАЦИЯ ПРОГРАММЫ С ПОМОЩЬЮ ПУЛЬТА ОБУЧЕНИЯ

1. Включите пульт обучения.
2. Нажмите клавишу MENUS (меню), чтобы отобразить экранное меню.
3. Выберите SELECT.

Еще один способ - можно также вызвать следующий экран выбора программы, нажав клавишу SELECT на пульте обучения.

```
SELECT                               JOINT 30%
                                     61276 bytes free
1  SAMPLE1      SAMPLE PROGRAM1
2  SAMPLE2      SAMPLE PROGRAM2
3  PROG001      PROGRAM001
4  PROG002      PROGRAM002

[TYPE]  CREATE  DELETE  MONITOR  [ATTR] >
COPY    DETAIL  LOAD    SAVE    PRINT  >
```

4. Нажмите клавишу F2 [CREATE] (создать). Отображается экран регистрации программы.

```
JOINT 30%
1  Words
2  Upper Case
3  Lower Case
4  Options      ---Insert---
SELECT
---Create Teach Pendant Program---
Program Name   [          ]
Sub type       [          ]
              ---End---
Enter program name
PRG    MAIN  SUB    TEST
```

5. При помощи клавиш курсора выберите способ ввода имени программы (слова или буквенные символы).

```
JOINT 30%
1 Words
2 Upper Case
3 Lower Case
4 Options ---Insert---
SELECT
---Create Teach Pendant Program---
Program Name [ ]
---End---
Enter program name
abcdef ghijkl mnopqr stuvwxyz_@* >
```

6. Введите имя программы, нажимая функциональные клавиши, соответствующие символам в имени программы. Отображаемое меню функциональных клавиш зависит от способа, выбранного в пункте 4.

Например, при вводе буквенных символов, нажмите функциональную клавишу, соответствующую нужному символу несколько раз, пока символ не отобразится в поле имени программы; то есть, если необходимо ввести P, нажмите функциональную клавишу F4 четыре раза. Нажмите клавишу NEXT (далее), чтобы переместить курсор вправо на один символ. Повторяйте эту процедуру, пока имя программы не будет полностью введено.

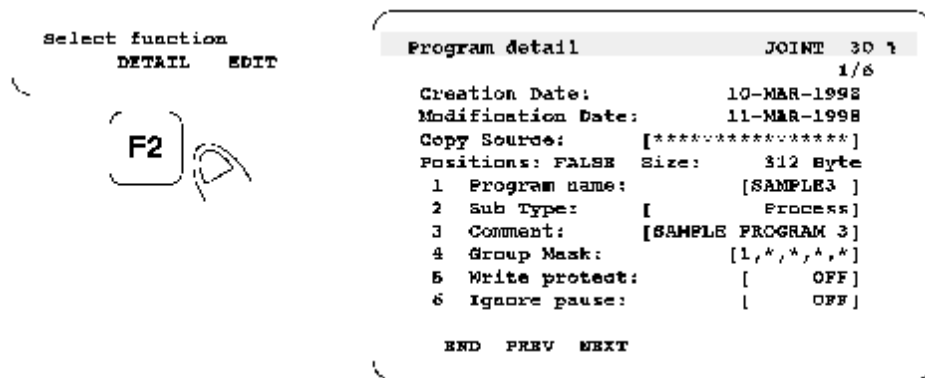
7. После ввода имени программы нажмите клавишу ENTER.

```
SELECT
---Create Teach Pendant
Program Name: [SAMPLE3]
ENTER

Select JOINT 30%
1 Jobs 5
2 Processes 6
3 Macro 7
4 $
Select
---Create Teach Pendant Program---
Program Name: [SAMPLE3]
Sub type [ ]
---End---
Select Sub type
```

8. Чтобы внести дополнительные данные в зарегистрированную программу, нажмите клавишу F3 (EDIT) (редактировать) или ENTER. Отображается экран редактирования программы для зарегистрированной программы.

9. Для ввода дополнительной информации программы нажмите клавишу F2 (DETAIL) (данные) (или клавишу ENTER). Отображается экран информации программы.



10. При необходимости задайте следующие пункты информации программы:

- Для изменения имени программы переместите курсор в поле настройки, измените имя программы, затем нажмите клавишу ENTER.
- Для изменения подтип , нажмите клавишу F4 [CHOICE] (выбор), чтобы отобразить меню подтипа. Затем выберите None (нет), Job (задание), Process (процесс) или Macro (макрос).

Подтип (sub type) используется для указания типа программы. Доступны следующие подтипы:

- Job, JB (задание). Обозначает основную программу, которую можно запустить на устройстве, например на пульте обучения.

В основной программе вызываются для выполнения программы процессов.

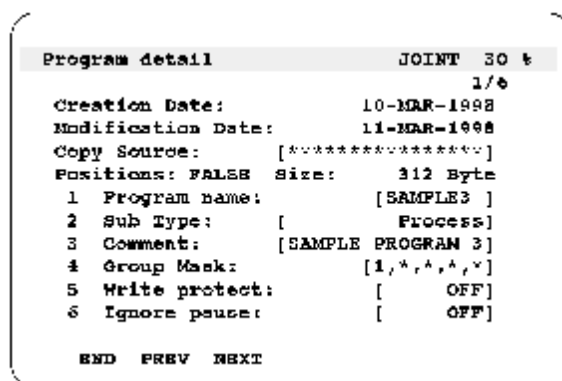
- Process, PR (процесс). Устанавливается по умолчанию. Обозначает подпрограмму, которая вызывается программой задания для выполнения определенного задания.

- Macro (MR) (макрос). Обозначает программу для выполнения макрокоманд. Для программы, зарегистрированной на экране настройки макрокоманд, автоматически устанавливается подтип MR.

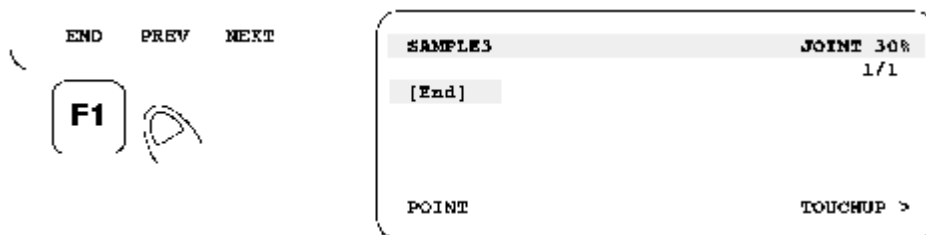
- Для ввода комментариев переместите курсор в поле настройки, введите комментарии, затем нажмите клавишу ENTER. Комментарий к программе может содержать от одного до шестнадцати символов.

Остальные настройки нам не нужны.

Для возврата в экран списка несколько раз нажмите клавишу PREV (назад), пока не появится экран списка.



11. После ввода пунктов информации программы нажмите клавишу F1 (END) (конец). Отображается экран редактирования программы для зарегистрированной программы.



3.2. КОМАНДЫ РЕГИСТРА

Регистр - это переменная для хранения целого числа или десятичной дроби. Система поддерживает двести регистров.

Имеются следующие команды регистра:

- Команды регистра
- Команды регистра положения
- Команды оси регистра положения

Instruction		JOINT 30%
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM		

Команды регистра допускают выполнение некоторых арифметических операций. В данном пособии они не рассматриваются.

Формат команды:

$$R[i] = (\text{значение})$$

Команда $R[i] = (\text{значение})$ загружает значение в заданный регистр.

i - номер регистра (1-32)

Значением, в частности, может быть:

- константа
- значение элемента регистра положения [i, j]
- значение $R[i]$
- аналоговый входной сигнал
- аналоговый выходной сигнал
- цифровой входной сигнал системы
- цифровой выходной сигнал системы
- цифровой входной сигнал робота
- цифровой выходной сигнал робота
- входной сигнал периферийного устройства
- выходной сигнал периферийного устройства

Ввод команды регистра в режиме обучения

1. Включите пульт обучения.
2. Выберите экран редактирования программы (**MENU, SELECT, имя программы**).

```
PROGRAM1 JOINT 30t
2/2
1: J P[1] 100t FINE
[End]

[INST] [EDCMD] >
```

3. Переместите курсор в пункт END (конец).
4. Нажмите клавишу F1 INST. При этом отображается меню команды управления.

[INST]

F1

Instruction		JOINT 30t
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM		

5. Выберите REGISTERS.

Instruction	
1 Registers	5
2 I/O	6
3 IF/SELECT	7
4 WAIT	8

ENTER

6. Выберите действие над регистром

```

REGISTER statement                                JOINT 30%
1 ...=...                    5 ...=.../...
2 ...=...+...                6 ...=...DIV...
3 ...=...-...                7 ...=...MOV...
4 ...=...*...                8
PROGRAM1

```

```

REGISTER statement                                JOINT 30%
1 R[ ]                    5
2 PL[ ]                   6
3 PR[ ]                   7
4 PR[i,j]                 8
PROGRAM1

```

2/3

```

2: ...=...+...
[End]

```

```

REGISTER statement                                JOINT 30 %
1 R[ ]                    5 RO[ ]
2 Constant                6 RI[ ]
3 DO[ ]                   7 GO[ ]
4 DI[ ]                   8 ---next page---
PRG1

```

2/2

```

2: R[1]=...+...
[End]

```

```

REGISTER statement                                JOINT 30%
1 R[ ]                    5 RO[ ]
2 Constant                6 RI[ ]
3 DO[ ]                   7 GO[ ]
4 DI[ ]                   8 ---next page---
PROGRAM1

```

2/3

```

2: R[1]=R[1]+...
[End]

```

```

PROGRAM11                                JOINT 30%

```

3/3

```

1: J P[1] 100% FINE
2: R[1]=R[1]+1
[End]

[INST]                                [EDCMD] >

```

На примере показано, что значение регистра R[1] увеличено на единицу.

3.3. КОМАНДЫ ПЕРЕМЕЩЕНИЯ

Команда перемещения вызывает перемещение инструмента робота в заданную точку в пределах рабочей зоны с заданной скоростью подачи и в заданном режиме перемещения. В команде перемещения должны быть заданы перечисленные ниже элементы.

Формат команды

Формат движения	Формат данных позиционирования	Данные позиционирования	Скорость перемещения	Траектория позиционирования
-----------------	--------------------------------	-------------------------	----------------------	-----------------------------

3.3.1. Формат движения

Для формата движения указывается траектория перемещения в заданное положение.

Возможны три варианта:

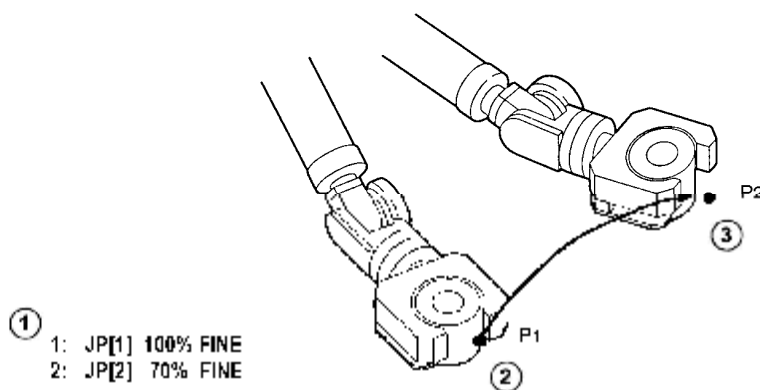
- движение шарнира (J),
- линейное перемещение (в том числе поворот)(L),
- круговое движение (C).

Движение шарнира J

Режим движения шарнира – основной режим перемещения робота в заданное положение.

Робот приходит в ускоренное движение вдоль всех или вокруг всех осей, перемещается с заданной скоростью подачи, замедляется и останавливается одновременно по всем осям.

Траектория перемещения обычно нелинейна. Формат движения указывается для определения конечной точки. Скорость подачи движения шарнира указывается в виде процента максимальной скорости. Ориентация перемещаемого инструмента не контролируется.



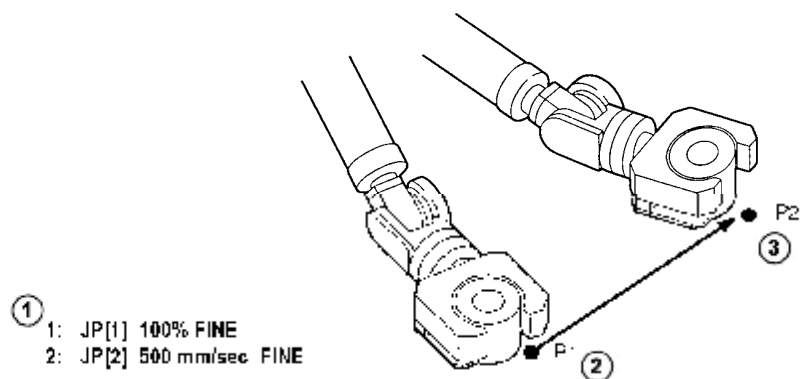
Линейное перемещение L

В режиме линейного перемещения контролируется траектория движения вершины инструмента из начальной точки в конечную точку; вершина ин-

инструмента перемещается линейно. Формат движения указывается для определения конечной точки.

При указании линейной скорости подачи следует выбрать единицу измерения: мм/с, см/мин и дюймы/мин.

Ориентация перемещаемого инструмента контролируется путем вычитания ориентации в начальной точке из ориентации в конечной точке.

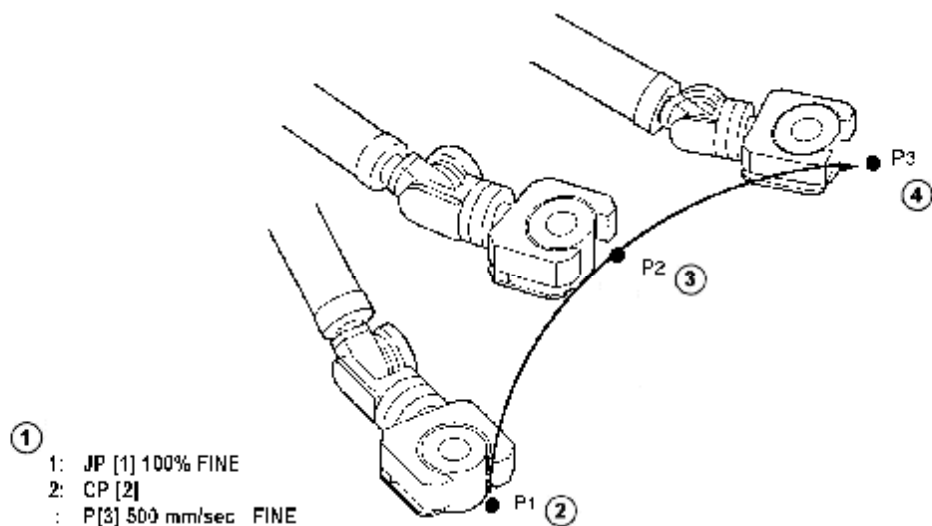


Круговое движение C

В режиме кругового движения контролируется траектория движения вершины инструмента из начальной точки в конечную точку через промежуточную точку. Промежуточная и конечная точки определяются в одной команде.

При указании круговой скорости подачи следует выбрать единицу измерения: мм/с, см/мин или дюймы/мин.

Ориентация перемещаемого инструмента контролируется путем вычитания ориентации в начальной точке из ориентации в конечной точке.



В данном пособии рассматривается формат движения шарнира J.

3.3.2. Формат данных позиционирования

В команде перемещения данные позиционирования представлены переменной положение переменной ($P[i]$) или регистром положения ($PR[i]$).

3.3.3. Данные позиционирования

Данные позиционирования хранятся в переменной или регистре положения и определяют ориентацию робота.

Данные позиционирования записываются в программу одновременно с обучением команде перемещения.

Данные позиционирования подразделяются на два типа. К одному типу относятся координаты шарниров в системе координат шарниров. К другому типу относятся декартовы координаты, описывающие положение и ориентацию инструмента в рабочем пространстве.

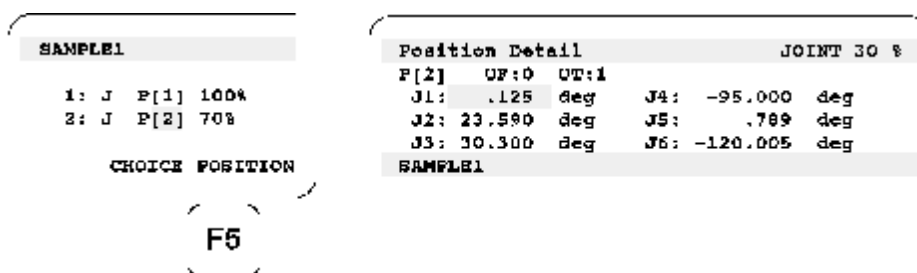
В данном пособии рассматривается система координат шарниров .

Координаты шарнира

Данные позиционирования определяются с помощью угловых смещений относительно системы координат шарнира.

$(J_1, J_2, J_3, J_4, J_5, J_6)$
Основные оси Оси запястья

Подробные данные позиционирования отображаются, если нажать клавишу F5 [POSITION] (положение), предварительно выбрав переменную положения.



В команде перемещения данные позиционирования представлены переменной положения ($P[i]$) или регистром положения ($PR[ir]$). Обычно используется переменная положения.

Где i – номер положения;

ir – номер регистра положения.

Для обращения к переменной положения используется номер положения i . Номер положения автоматически присваивается каждый раз, когда происходит обучение команде перемещения и это отражается в программе. Например, сначала присваивается номер положения $P[1]$, затем $P[2]$ и т.д.

Регистр положения служит универсальным регистром для хранения данных позиционирования (См. ниже).

Номер положения и номер регистра положения можно дополнить комментарием длиной до 16 символов. Чтобы добавить комментарий, нажмите клавишу ENTER, когда курсор находится на номере положения. Например,

4: J P[11: **APPROACH POS**] 30% FINE

3.3.4. Скорость перемещения шарнира

Для движения шарнира (J) скорость перемещения указывается в процентах от максимально допустимой в пределах от 1% до 100%. Например,

J P[1] 50% FINE

Для указания скорости в других системах (L, C) возможны и другие способы. Но об этом позже.

3.3.5. Траектория позиционирования

Существует два режима траекторий позиционирования:

- траектория позиционирования FINE,
- траектория позиционирования CNT.

Траектория позиционирования FINE

J P[i] 50% **FINE**

Если задана траектория позиционирования FINE, робот останавливается в конечной точке, прежде чем перемещаться к следующей конечной точке.

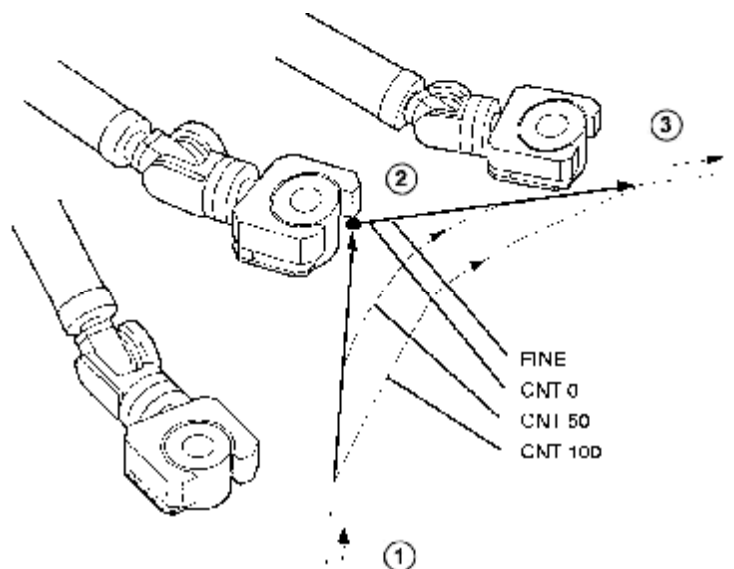
Траектория позиционирования CNT

J P[i] 50% **CNT50**

Если задана траектория CNT, робот приближается к промежуточной точке позиционирования, но не останавливается в ней и перемещается к следующей точке. Можно указать, насколько близко робот должен приблизиться к промежуточной точке, указав значение в пределах от 0 до 100.

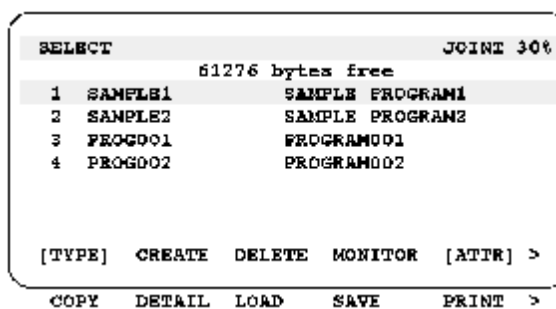
Если указан 0, робот перемещается к конечной точке максимально приближаясь к промежуточным точкам позиционирования, но не останавливается в них.

Если указано 100, робот перемещается к конечной точке по самой удаленной траектории.



3.3.6. Ввод команды перемещения в режиме обучения

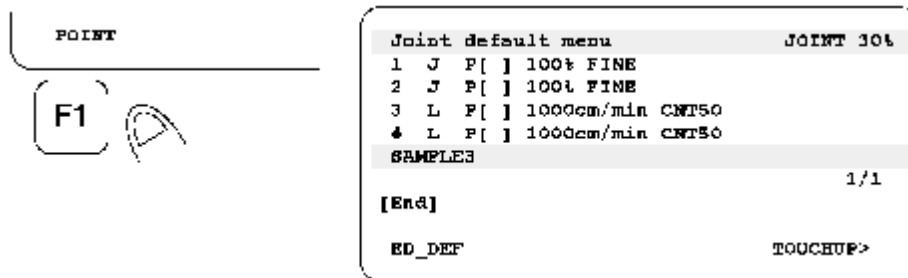
1. Нажмите клавишу SELECT на пульте обучения



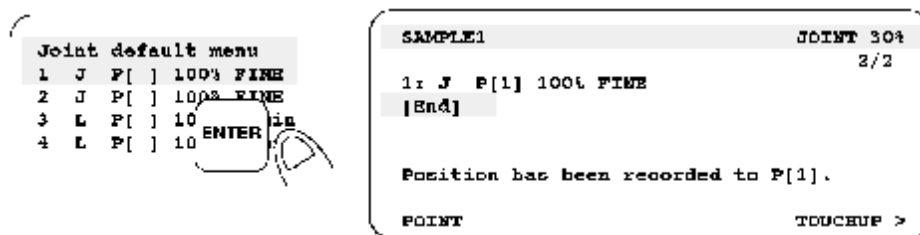
2. С помощью кнопок управления курсором выберите программу, которую хотите отредактировать, или зарегистрируйте новую. (См. выше)



3. Переместите робота в ручном режиме в нужное положение в рабочей зоне.
4. Переместите курсор в пункт END (конец).
5. Нажмите клавишу F1 [POINT], чтобы отобразить меню стандартных команд перемещения.



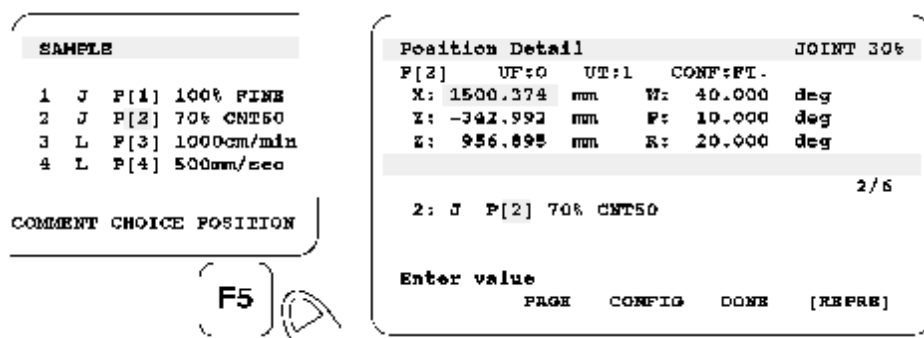
6. Выберите стандартную команду перемещения, вводимую в режиме обучения, нажмите клавишу ENTER и укажите нужную команду перемещения. При этом данные позиционирования вводятся автоматически.



7. Повторите пункты с 3 по 6 для каждой команды перемещения, которая будет задана в программе.

8. Чтобы задать одну и ту же стандартную команду перемещения много раз, нажимая клавишу SHIFT, нажмите клавишу F1 [POINT]. Это добавляет ранее заданную команду перемещения к стандартной команде перемещения, выбранной в настоящий момент.

9. Для отображения информации данных позиционирования переместите курсор к нужной переменной положения, а затем нажмите клавишу F5 [POSITION] (положение). Будет отображен экран информации данных позиционирования.



3.3.7. Изменение данных позиционирования с помощью робота

1. Выберите программу, которая будет изменена (MENU, SELECT).
2. Включите пульт обучения.
3. Переместите курсор в номер строки, в которой отображается команда перемещения, которую необходимо изменить.

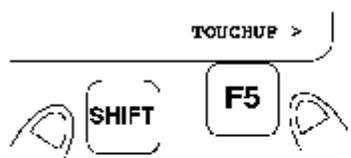
```

SAMPLE1                               JOINT 30%
                                       2/6
1 J P[1] 100% FINE
2 J P[2] 70% CNT50
3 L P[3] 1000cm/min CNT30
4 L P[4] 500mm/sec FINE
5 J P[1] 100% FINE
[End]

POINT                                TOUCHUP >

```

4. Переместите робот в новое положение. Нажмите F5 [TOUCHUP] (касание) при нажатой клавише SHIFT. Новое положение записано.



```

SAMPLE1                               JOINT 30%
                                       2/6
1 J P[1] 100% FINE
2 J P[2] 70% CNT50
3 L P[3] 1000cm/min CNT30
4 L P[4] 500mm/sec FINE
5 J P[1] 100% FINE
[End]
Position has been recorded to P[2].
POINT                                TOUCHUP >

```

3.3.8. Изменение данных позиционирования с помощью экрана информации

1. Выберите программу, которая будет изменена (MENU, SELECT).
2. Включите пульт обучения.
3. Переместите курсор в номер строки, в которой отображается команда перемещения, которую необходимо изменить.

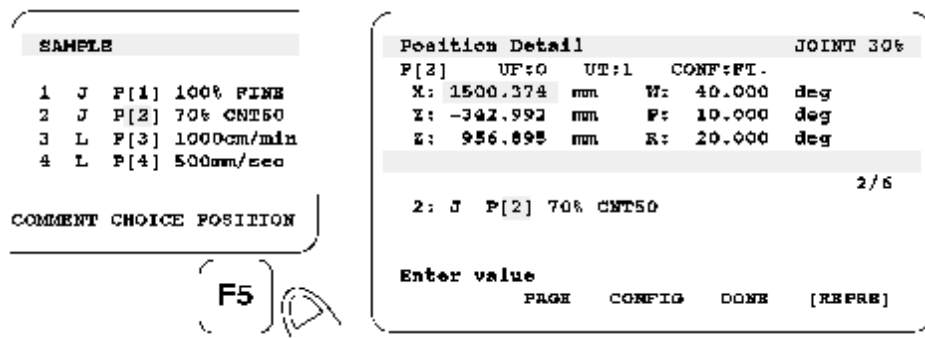
```

SAMPLE1                               JOINT 30%
                                       2/6
1 J P[1] 100% FINE
2 J P[2] 70% CNT50
3 L P[3] 1000cm/min CNT30
4 L P[4] 500mm/sec FINE
5 J P[1] 100% FINE
[End]

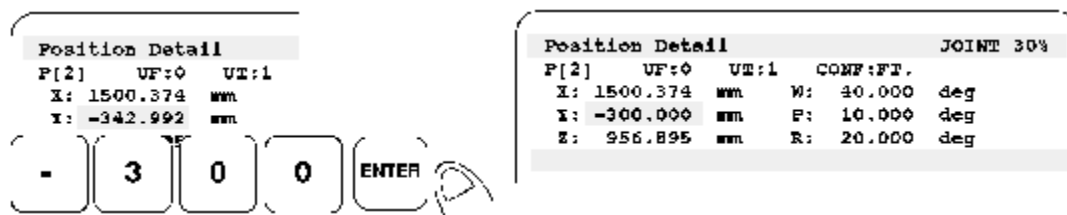
POINT                                TOUCHUP >

```

4. Для отображения информации данных позиционирования переместите курсор к нужной переменной положения, а затем нажмите клавишу F5 [POSITION] (положение). Будет отображен экран информации данных позиционирования.



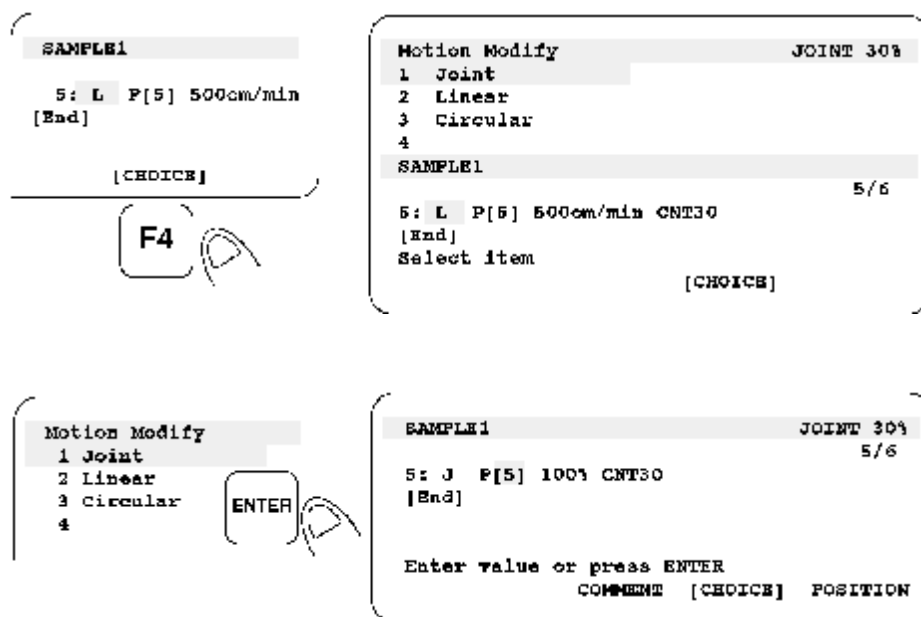
5. Для изменения положения переместите курсор к координатам для каждой оси и введите новые координаты.



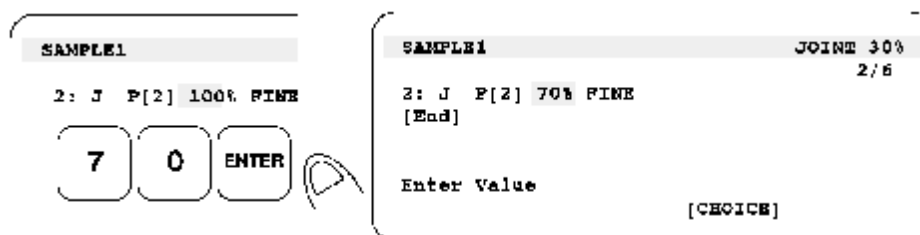
3.3.9. Изменение команды перемещения

1. Переместите курсор в элемент команды перемещения, которую необходимо изменить.
2. Для отображения подменю элементов команды нажмите клавишу F4 [CHOICE], затем выберите в подменю элемент команды, который необходимо изменить.

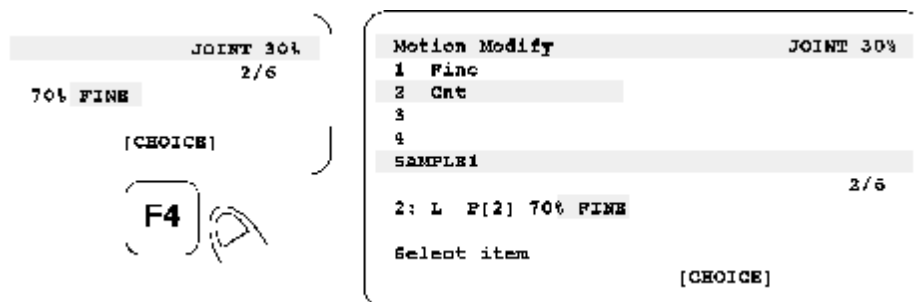
На следующих экранах показано изменение типа перемещения с линейного перемещения на движение шарнира:



3. Изменение скорости подачи.



4. Изменение типа позиционирования.



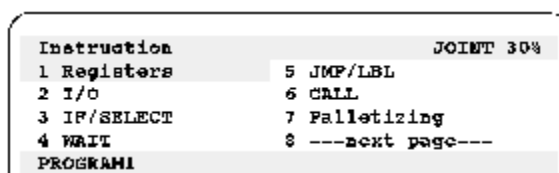
3.3.10. Позиционирование работа с помощью регистров положения

Регистр положения (**PR[i]**) - это переменная для хранения данных позиционирования. Имеется сто регистров положения.

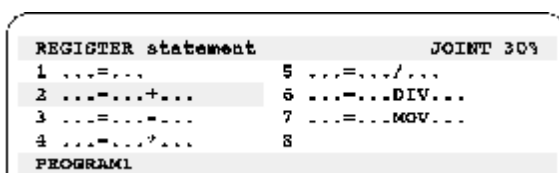
Для отображения и задания значений регистров используется экран регистров положения.

Ввод команды регистра положения в режиме обучения

1. Переместите курсор в пункт END (конец).
2. Нажмите клавишу F1 INST. При этом отображается меню команды управления.



3. Выберите REGISTERS.



4. Выберите PR[] Введите в режиме обучения команду, назначив декартовы координаты текущего положения регистру положения на следующих экранах.

```
REGISTER statement          JOINT 30 %
1 R[ ]                      5
2 PR[ ]                     6
3 PR[i,j]                   7
4                             8
PRG1
```

```
REGISTER statement          JOINT 30 %
1 LPOS                       5 UCOORD[ ]
2 JPOS                       6 PR[ ]
3 F[ ]                       7
4 UFRAME[ ]                  8
PRG1
2/3
2: PR[1]-...
[End]
Select item
[CHOICE]
```

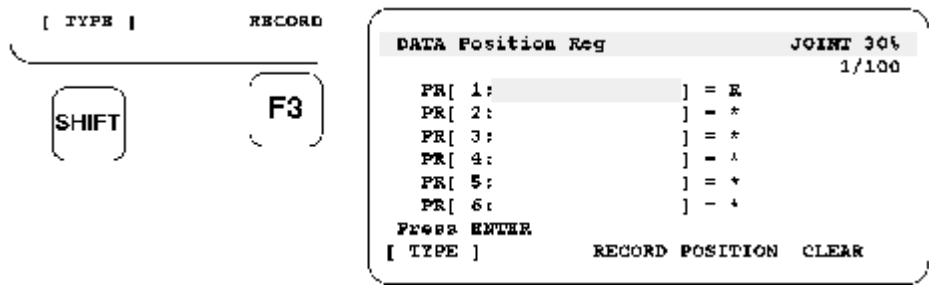
```
PROGRAM1                    JOINT 30%
2/3
2: PR[1]-LPOS
[End]
[INST]                      [EDCMD] >
```

Настройка регистров положения

1. Нажмите клавишу MENUS (меню), чтобы отобразить экранное меню.
2. Нажмите клавишу DATA (данные).
3. Нажмите F1 [TYPE] (тип) для отображения меню смены экранов.
4. Выберите Position Reg. Появится экран регистров положения.

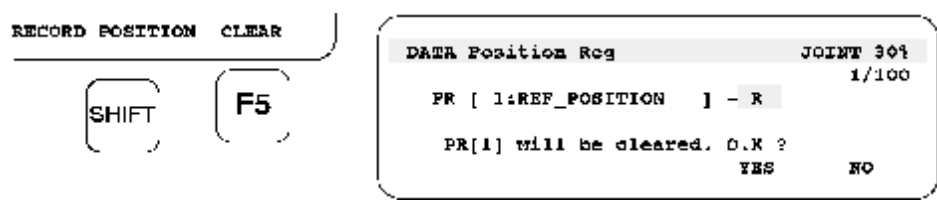
```
DATA Position Reg          JOINT 30%
1/100
PR[ 1: ] = *
PR[ 2: ] = *
PR[ 3: ] = *
PR[ 4: ] = *
PR[ 5: ] = *
PR[ 6: ] = *
Press ENTER
[ TYPE ]          RECORD POSITION CLEAR
```

5. Для ввода комментария :
 - a. Переместите курсор к нужному полю номера регистра положения, затем нажмите клавишу ENTER.
 - b. Выберите способ ввода символов.
 - c. С помощью функциональных клавиш введите комментарий.
 - d. После завершения ввода нажмите клавишу ENTER.
6. Для изменения значения регистра положения переместите курсор в поле значения регистра положения. Затем при нажатой клавише SHIFT нажмите F3 RECORD (запись).

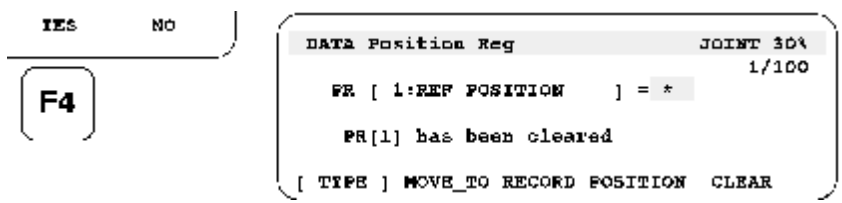


- Индикация R указывает, что в регистре положения уже хранится значение, введенное в режиме обучения.
- Звездочка (*) указывает, что значение не хранится.

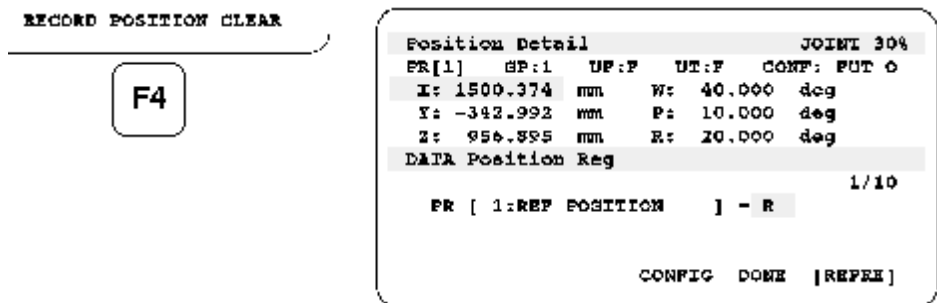
7. Для удаления данных позиционирования, загруженных в регистр положения нажмите F5 CLEAR (удалить) при нажатой клавише SHIFT.



8. Выберите YES (да). Данные позиционирования нужного регистра положения удалены.



9. Чтобы узнать текущие значения данных позиционирования, нажмите F4 POSITION (положение). Появится экран данных позиционирования. Для изменения значения переместите курсор к нужному полю, затем введите новое значение.



10. По окончании настройки нажмите F4 DONE (готово).

3.3.11. Команда коррекции ускорения

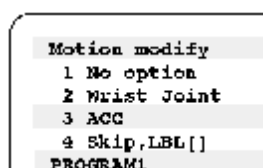
Эта команда указывает процентный коэффициент ускорения/замедления во время движения. Пределы ускорения от 0 до 150%.

При уменьшении коррекции ускорения время ускорения увеличивается (ускорение и замедление выполняются медленнее). Для выполнения потенциально опасных операций, таких как зачерпывание горячей воды, следует использовать значение АСС менее 100%. Для участков, на которых робот перемещается слишком медленно, задайте значение АСС больше 100%.

Формат команды

J P[1] 50% FINE ACC80

Для ввода команды коррекции ускорения установите курсор после команды перемещения и нажмите клавишу F4 [CHOICE] (выбор), чтобы открыть список дополнительных команд перемещения. Затем выберите нужную дополнительную команду перемещения АСС.



3.4. КОМАНДА ПЕРЕХОДА

Команды перехода служат для изменения последовательности выполнения команд в пределах одной программы.

Поддерживаются четыре типа команд перехода.

- команда определения метки,
- команда конца программы,
- команда безусловного перехода,
- команда условного перехода.

3.4.1. Команда определения метки

Команда метки (LBL[i]) используется для указания точки в программе, к которой должно перейти управление.

Для ввода команды метки в режиме обучения нажмите (MENU), затем клавишу F1 (INST), чтобы вызвать подменю. Выберите в подменю элемент команды управления JMP/LBL.

Instruction		JOINT 304
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM		

Формат команды

LBL [i: Comment]

I-номер метки (1—32786)

Comment-комментарий (1—16 символов)

Введите номер метки и, если захотите, комментарий. Чтобы добавить комментарий, переместите курсор на номер метки и нажмите клавишу ENTER.

После определения метки ее можно использовать для безусловного или условного перехода.

3.4.2. Команды безусловного перехода

Поддерживаются два типа команд безусловного перехода.

- Команда перехода вызывает передачу управления на указанную метку в пределах одной программы (JMP).
- Команда вызова подпрограммы (CALL).

1. Команда безусловного перехода JMP LBL[i] передает управление программы на заданную метку в пределах одной программы (I от 1 до 32767).

Формат команды

JMP LBL[i]

2. Команда вызова подпрограммы CALL (подпрограмма). Данное программное обеспечение не различает понятия программа и подпрограмма, т.е. программа может быть и подпрограммой и наоборот. Поэтому условимся, программная единица будет называться подпрограммой, если она вызывается из другой программы (подпрограммы). Программная единица будет называться программой, если она вызывается с помощью команды выполнения программы. (См. ниже).

Команда вызова подпрограммы CALL (подпрограмма) передает управление на первую строку другой программы (подпрограммы) для ее выполнения. Когда в вызванной программе выполняется команда конца программы (END), управление передается команде следующей сразу после команды вызова программы в вызывающей программе. Для ввода в режиме обучения команды вызова подпрограммы нажмите клавишу F1 (INST). Затем выберите в подменю элемент команды управления CALL.

Чтобы задать имя вызываемой программы, выберите его в автоматически отображаемом подменю или нажмите F5, STRINGS и введите символы.

Формат команды

CALL (подпрограмма)

Instruction		JOINT 303
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM		

3.4.3. Команда конца программы и подпрограммы

Этой командой завершается выполнение программы.

Если программа вызвана из другой программы, управление возвращается в основную программу на строку, следующую за строкой вызова подпрограммы. Команда **END (конец)** вставляется автоматически в конце программы и подпрограммы.

3.4.4. Использование аргументов в команде CALL

Можно задать аргумент для команды вызова программы и использовать его значение в подпрограмме. При помощи аргументов и регистров аргументов можно передавать данные только между двумя программами. Например,

```
CALL PROC_1 (1, AR[1], R[6])
          ↓   ↓   ↓
PROC 1 : AR[1], AR[2], AR[3]
```

Аргументы, которые можно использовать в вызывающей программе:

-- константа, например, 1, 3.5,

-- строка символов, например, «Парковка»,

- регистр аргумента, например, AR[3],
- регистр, например, R [6].

Пример вызова подпрограммы: CALL SUBPRG (1, R[3], AR[1])

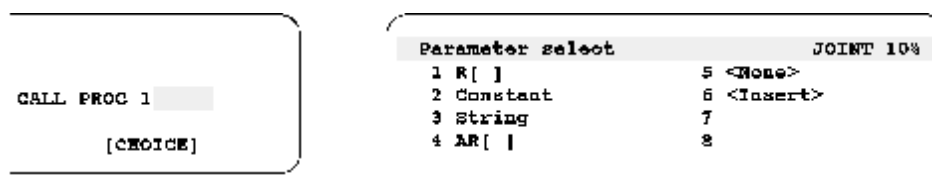
В вызываемой подпрограмме в качестве регистра аргумента удобно использовать в командах, в левой части которых используется регистр. Например,

```
R[1]=AR[2]
R[4]=AR[4]
```

3.4.5.Задание аргументов

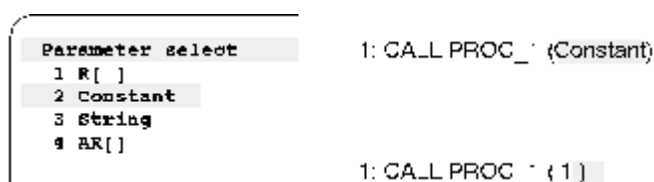
После указания команды вызова программы или макрокоманды курсор останавливается в конце строки. Если задавать аргументы не требуется, нажмите клавишу ENTER или клавишу со стрелкой вправо или вниз, чтобы переместить курсор на следующую строку.

Чтобы открыть подменю выбора аргументов, нажмите функциональную клавишу [CHOICE](выбор).



1. Задание аргументов типа константы.

Чтобы указать аргумент типа константы, нажмите функциональную клавишу [CHOICE] и выберите в подменю пункт «2 Constant» (константа).

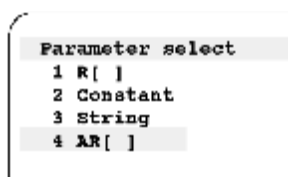


2. Задание аргументов символьного типа.

В данном пособии не рассматривается за ненадобностью.

3. Задание аргументов типа регистра аргумента.

Чтобы указать аргумент типа регистра аргумента, нажмите функциональную клавишу [CHOICE] и выберите в подменю пункт «AR[]»



Введите индекс. Например, 1: CALL PROC_1 (AR[1])

ПРИМЕЧАНИЯ по применению команд регистров аргумента .

Значения и количество аргументов, которые требуется передать между вызывающей и вызываемой программами, не проверяется при их задании. Если аргумент задан или используется неправильно, происходит ошибка в строке, в которой во время выполнения программы обнаруживается конфликт.

СОВЕТЫ.

- Убедитесь, что число аргументов, заданных в основной программе, равняется числу аргументов, используемых в подпрограмме. При этом, если аргументы, указанные в основной программе, не используются в подпрограмме, ошибки не происходит.
- Убедитесь, что содержимое аргументов, заданных в основной программе, соответствует типам команд в подпрограмме, использующих эти аргументы.
- Убедитесь, что индексы и значения указанных аргументов заданы правильно. Например,
1 CALL PROG_1 (Constant) Происходит ошибка, так как значение не инициализировано.
1 CALL PROG_1 (R[...]) Индекс не инициализирован
При выполнении строк, содержащих эти команды, произойдет ошибка с кодом «INTP-201 Unspecified Statement» (не указан оператор).

3.4.6. Команды условного перехода

Команда условного перехода вызывает переход из одного места программы в другое при удовлетворении некоторого условия.

Имеются два типа команд условного перехода.

- Команда условного сравнения: вызывает передачу управления на указанную метку или в указанную программу при удовлетворении некоторого условия. Имеются команды условного сравнения регистра и условного сравнения ввода-вывода.
- Команда условного сравнения сигналов ввода вывода: вызывает передачу управления на заданную команду перехода или вызова подпрограммы в соответствии со значением регистра.

Для ввода в режиме обучения команды условного перехода нажмите клавишу F1 (INST), чтобы вызвать подменю. Затем выберите в подменю элемент команды управления IF/SELECT.

Instruction	JOINT 308
1 Registers	5 JMP/LBL
2 I/O	6 CALL
3 IF/SELECT	7 Palletizing
4 WAIT	8 ---next page---
PROGRAM	

1. Команда условного сравнения регистра IF R[i] (оператор) (значение) (обработка) сравнивает значение, хранящееся в регистре, с другим значением. Если условие сравнения удовлетворено, выполняет обработку (JMP или CALL).

Формат команды

IF R[i] (оператор) (значение) (обработка)

Оператор – символы:(>), (>=), (=), (<), (<=), (<>).

Значение – константа, регистр R[i].

Обработка-- JMP LBL[i], CALL (подпрограмма).

Если содержимое регистра сравнивается с действительным значением при помощи оператора «=», оно не всегда может соответствовать действительному значению из-за ошибок округления. При операциях с действительными (аналоговыми) величинами следует использовать другой символ, например, «<>».

2. Команда условного сравнения ввода-вывода периферийных устройств IF (сигнал) (оператор) (значение) (обработка) сравнивает значение сигнала ввода-вывода с другим значением. Если условие сравнения удовлетворено, выполняется заданная обработка.

Формат команды для обработки цифровых сигналов управления

IF (сигнал) (оператор) (значение) (обработка)

Сигнал-- цифровой ввод DI[i], цифровой вывод DO[i].

Оператор – символы (=), (<>).

Значение--ON, OFF, DI[i], DO[i].

Обработка-- JMP LBL[i], CALL (подпрограмма).

Формат команды для обработки аналоговых сигналов управления

IF (сигнал) (оператор) (значение) (обработка)

Сигнал-- аналоговый ввод AI[i], аналоговый вывод AO[i].

Оператор – символы:(>), (>=), (=), (<), (<=), (<>).

Значение – константа, регистр R[i].

Обработка-- JMP LBL[i], CALL (подпрограмма).

Внимание! Для аналоговых сигналов знак «=» лучше не применять.

3.4.7. Команда условного выбора

Команда условного выбора состоит из нескольких команд сравнения регистров. Команда условного выбора сравнивает значение регистра с одним или несколькими значениями, а затем выбирает оператор, удовлетворяющий условию сравнения.

- Если значение указанного регистра совпадает с одним значением, выполняется команда перехода или вызова подпрограммы, соответствующая этому значению.
- Если значение заданного регистра не совпадает ни с одним значением, выполняется команда перехода или вызова подпрограммы, соответствующей условию ELSE.

```

SELECT R [ i ] = (value) (Processing)
Register number | = (value) (Processing)
(1 to 32)       | = (value) (Processing)
                | ELSE
                |   Constant | JMP LBL [ i ]
                |   R [ i ]  | CALL ( Program )

```

① Example

```

11: SELECT R[1] = 1, JMP LBL[1]
12:   = 2, JMP LBL[2]
13:   = 3, JMP LBL[2]
14:   = 4, JMP LBL[2]
15:   ELSE, CALL SUB2

```

Формат команды

**SELECT R[1] = (значение) (обработка) = (значение) (обработка)
= (значение) (обработка)
ELSE (обработка)**

3.5. КОМАНДЫ ОЖИДАНИЯ

Команда ожидания используется для остановки выполнения программы на заданный период времени или пока не будет удовлетворено условие. Доступны два типа команд ожидания.

- Команда ожидания с ограничением по времени. Задерживает выполнение программы на заданный промежуток времени.
- Условная команда ожидания выполнения условия. Задерживает выполнение программы, пока не будет удовлетворено заданное условие или не истечет заданный период времени.

Для ввода команды ожидания в режиме обучения нажмите клавишу F1 (INST), чтобы вызвать подменю. Затем выберите в подменю элемент команды управления WAIT.

Instruction	JOINT 301
1 Registers	5 JMP/LBL
2 I/O	6 CALL
3 IF/SELECT	7 Falltizing
4 WAIT	8 ---next page---
PROGRAM	

3.5.1. Команда ожидания с ограничением по времени

Команда ожидания с ограничением по времени **WAIT (время)** задерживает выполнение программы на заданный период времени (в секундах) с дискретностью 0.1С.

Формат команды

WAIT (время)

Время можно задать в виде константы, например, WAIT 10.5sec или может содержаться в регистре R[i].

3.5.2. Команда ожидания выполнения условия

Команда ожидания выполнения условия **WAIT (условие) (обработка)** задерживает выполнение программы, пока не будет удовлетворено заданное условие.

Если обработка не задана, выполнение программы останавливается, пока заданное условие не будет удовлетворено. Когда условие выполнится, управление переходит к следующей строке программы.

1. *Условная команда ожидания регистра* **WAIT R[i] (оператор) (константа/ R[i])** сравнивает значение регистра с другим значением и задерживает выполнение программы, пока не удовлетворено условие сравнения.

Формат команды

WAIT R[i] (оператор) (константа/ R[i])

Оператор – символы:(>), (>=), (=), (<), (<=), (<>).

2. *Условная команда ожидания ввода-вывода с периферийных устройств* **WAIT (сигнал) (оператор) (константа/ R[i])** сравнивает значение сигнала ввода-вывода с другим значением и задерживает выполнение программы, пока не удовлетворено условие сравнения.

Формат команды для обработки аналоговых сигналов управления

WAIT (AO/A1) (оператор) (константа/ R[i])

Оператор – символы:(>), (>=), (=), (<), (<=), (<>).

Внимание! Для аналоговых сигналов знак «=>» лучше не применять.

Формат команды для обработки цифровых сигналов управления

WAIT (DO/D1) (= / <>) (значение)

Значение--ON, OFF, DI[i], DO[i].

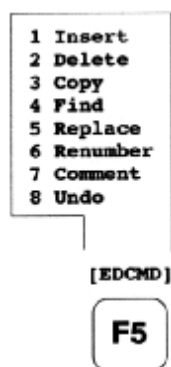
-- Off- (выкл.-): задний фронт сигнала принимается за условие обнаружения. Условие не удовлетворено, пока сигнал остается выключенным. Условие обнаружения удовлетворено, если сигнал изменяется из состояния «включено» в состояние «выключено».

-- On+ (вкл.+): передний фронт сигнала принимается за условие обнаружения. Условие не удовлетворено, пока сигнал остается включенным. Условие обнаружения удовлетворено, если сигнал изменяется из состояния «выключено» в состояние «включено».

4. КОМАНДЫ РЕДАКТИРОВАНИЯ ПРОГРАММЫ

Команды редактирования программы используются для редактирования программы.

Нажмите NEXT (далее), > для вызова F5 EDCMD, чтобы отобразить меню команд редактирования программы и выберите из меню нужную команду редактирования программы.



Insert (вставка). Для вставки заданного количества пустых строк между имеющимися строками программы. При вставке пустых строк строки программы перенумеровываются.

Delete (удалить). Удаляет последовательность команд из программы. После того, как команды удалены, строки программы перенумеровываются.

Copy (копировать). Копирует последовательность команд и вставляет группу команд в другую точку в программе. При копировании последовательности команд группа команд выбирается и записывается в память.

После копирования последовательности команд ее можно вставлять в другие точки в программе сколько угодно раз.

Find (найти). Для поиска заданного элемента команды программы. Для быстрого поиска заданного элемента в длинной программе.

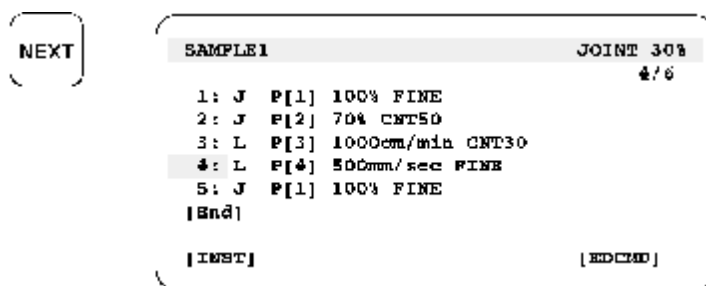
Replace (заменить). Для замены элемента заданной команды программы другим элементом. Эта команда используется, например, при изменении данных настройки программы. (Например, если изменяется назначение ввода-вывода и DO[1] необходимо переключить на DO[2] в программе).

Undo (отменить). Операции редактирования программы, такие как изменение команды, вставка строки и удаление строки можно отменить для возврата в состояние, существовавшее до того, как были выполнены эти операции редактирования. Если операция отмены выполняется при редактировании строки программы, отменяются все операции, выполненные в этой строке. Например, если строка была вставлена или удалена, восстанавливается состояние, которое существовало до операции вставки или удаления.

Если за операцией отмены немедленно выполняется другая операция отмены, то восстанавливается состояние, которое существовало до того, как была выполнена первая операция отмены.

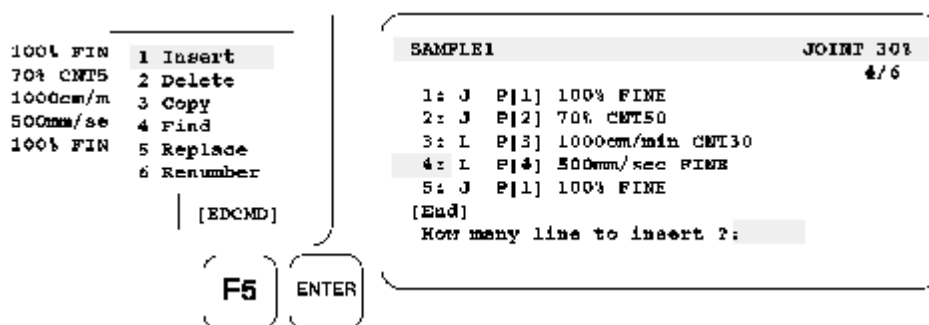
4.1. ВСТАВКА ПУСТЫХ СТРОК

1. Нажмите NEXT (далее), > для вызова F5 EDCMD.



2. Нажмите клавишу F5 [EDCMD]. Отображается меню команд редактирования.

3. Выберите Insert (вставить).



В примере ниже между 3 и 4 строкой вставляются две пустые строки.

4. Переместите курсор в строку, куда должны быть вставлены команды.

В этом примере переместите курсор в 4 строку.

5. Введите количество пустых строк, которые необходимо вставить (две), и нажмите клавишу ENTER.

2 ENTER

```
SAMPLE1 JOINT 30%
4/5
1: J P[1] 100% FINE
2: J P[2] 70% CNT50
3: L P[3] 1000cm/min CNT30
4:
5:
6: L P[4] 500mm/sec FINE
5: J P[1] 100% FINE
[INST] [EDCMD] >
```

Две пустые строки будут вставлены в программу, а все последующие строки в программе будут перенумерованы.

4.2. УДАЛЕНИЕ КОМАНД

1. Переместите курсор в левую часть строки, в которой находится удаляемая команда. (Задайте строку, которая будет удалена, с помощью курсора.)
2. Нажмите NEXT (далее), > для вызова F5 EDCMD.

NEXT

```
SAMPLE1 JOINT 30%
4/6
1: J P[1] 100% FINE
2: J P[2] 70% CNT50
3: L P[3] 1000cm/min CNT30
4: L P[4] 500mm/sec FINE
5: J P[1] 100% FINE
[End]
[INST] [EDCMD]
```


3. Нажмите клавишу F5 [EDCMD], чтобы вызвать меню команд редактирования.
4. Выберите Delete (удалить).

```
100% FIN 1 Insert
70% CNT5 2 Delete
1000cm/m 3 Copy
500mm/se 4 Find
100% FIN 5 Replace
          6 Renumber
          [EDCMD]
```

```
SAMPLE1 JOINT 30 %
4/6
1:J P[1] 100% FINE
2:J P[2] 70% CNT50
3:L P[3] 1000cm/min CNT30
4:L P[4] 500mm/sec FINE
5:J P[1] 100% FINE
[End]
Delete line(s) ? YES NO
F5 ENTER
```

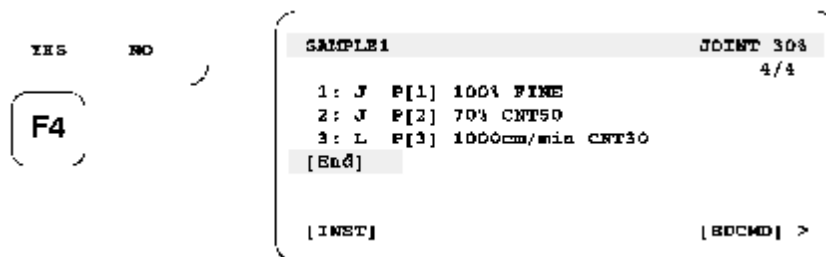
5. При помощи клавиш курсора (вверх и вниз) задайте диапазон командных строк, которые будут удалены.

```
3: L P[3] 1000cm/
4: L P[4] 500mm/s
5: J P[1] 100% FI
[End]
```



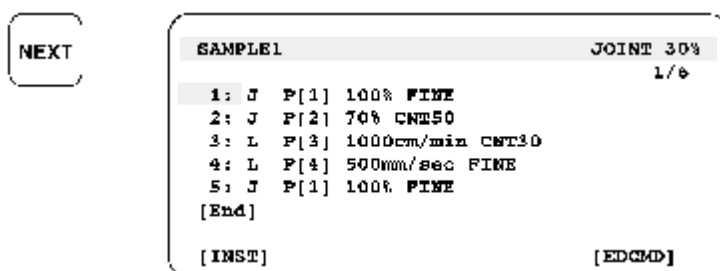
6. Для отмены удаления выбранной строки нажмите клавишу F5 (NO) (нет).

Для удаления выбранных строк нажмите клавишу F4 (YES) (да).



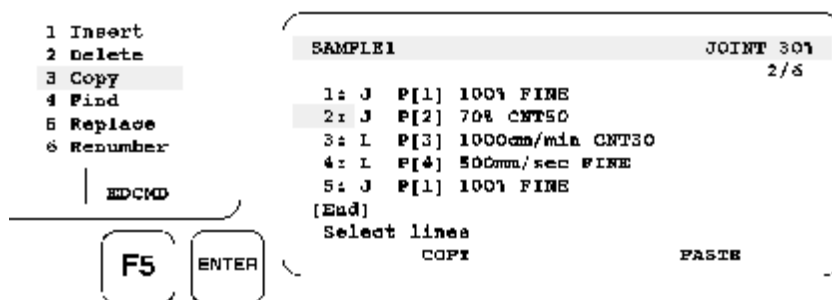
4.3. КОПИРОВАНИЕ И ВСТАВКА КОМАНД

1. Нажмите NEXT (далее), > пока не появится F5 EDCMD.

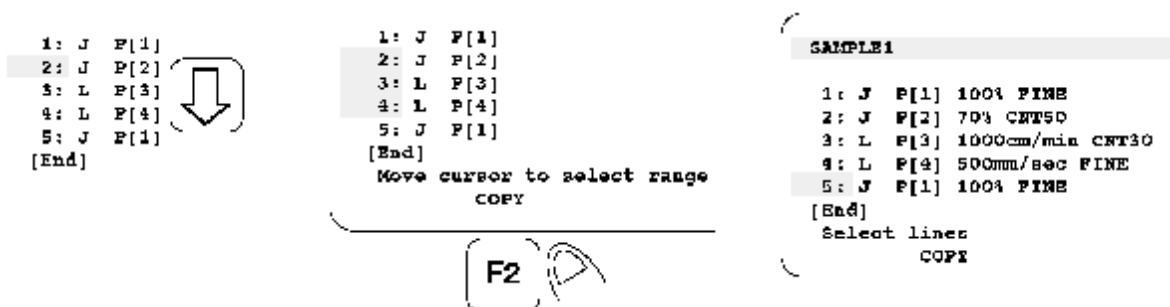


2. Нажмите клавишу F5 [EDCMD]. Отображается меню команд редактирования.

3. Выберите 3 Copy (копировать). На следующих экранах показано копирование строк со 2 по 4 в строки с 5 по 7.

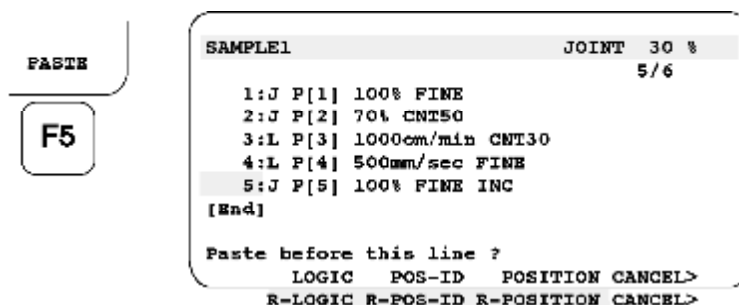


4. Выберите диапазон строк, которые будут скопированы

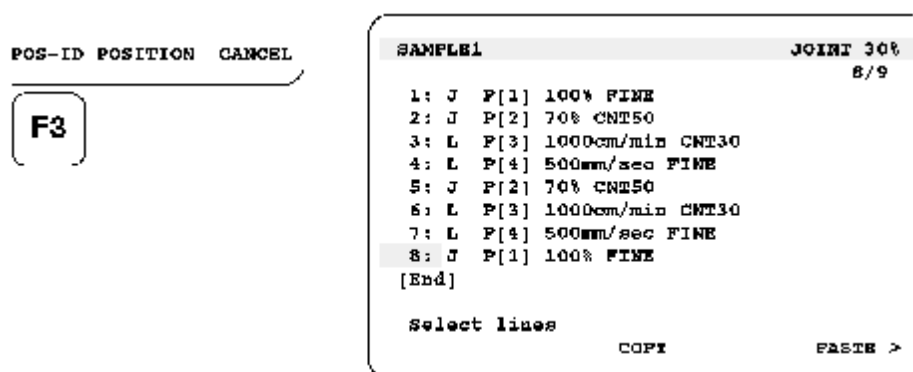


В результате выполнения вышеописанных пунктов, выбранные команды (строки с 2 по 4 в этом примере) были скопированы в память.

5. Решите, куда необходимо хотите вставить данные, скопированные в память.



6. Выберите способ копирования и вставки (копирования с оригинала).



7. При повторе действий вышеуказанных пунктов с 5 по 6 одну и ту же группу команд можно вставить неограниченное количество раз в любой точке в программе.

8. Для завершения вставки команд нажмите клавишу PREV.

5. ПРОГРАММИРОВАНИЕ ПЕРЕФЕЙНЫХ УСТРОЙСТВ

Для управления периферийными устройствами с данного контроллера служат команды ввода-вывода. К периферийным устройствам относится также инструмент устанавливаемый на ось J6. К командам управления периферийными устройствами относятся:

- команда (цифрового) ввода-вывода работа
- команда цифрового ввода-вывода (системы)
- команда аналогового ввода-вывода (системы)

Для доступа к командам управления необходимо из главного меню вызвать подменю, нажав F1 и клавишами управления курсором выберите пункт I/O.

Instruction		JOINT 30%
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM		

5.1. КОМАНДЫ ЦИФРОВОГО ВВОДА-ВЫВОДА РОБОТА

Входной сигнал робота (RI) и выходной сигнал робота (RO) - это сигналы ввода-вывода, которыми может управлять пользователь.

1. Команда $R[i] = RI[i]$ загружает состояние входного сигнала робота (включено = 1, выключено = 0) в заданный регистр.

$$R[i] = RI[i]$$

2. Команда $RO[i] = ON/OFF$ включает (ON) или выключает (OFF) заданный выходной сигнал робота.

$$RO[i] = ON/OFF$$

3. Команда $RO[i] = PULSE, [длительность]$ инвертирует текущее состояние указанного выхода робота на заданное время (от 0.1 до 25.5C).

$$RO[i] = PULSE, [длительность]$$

4. Команда $RO[i]=R[i]$ включает или выключает заданный выходной сигнал робота в соответствии со значением заданного регистра. Если заданный регистр имеет значение 0, выходной сигнал робота выключается. Если заданный регистр имеет значение, отличное от 0, выходной сигнал робота включается.

$$RO[i] = R[i]$$

5.2. КОМАНДЫ ЦИФРОВОГО ВВОДА-ВЫВОДА СИСТЕМЫ

Цифровой входной сигнал (DI) и цифровой выходной сигнал (DO) - это сигналы ввода-вывода, которыми может управлять пользователь.

1. Команда $R[i] = DI[i]$ загружает состояние цифрового входного сигнала (включено = 1, выключено = 0) в заданный регистр.

$$R[i] = DI[i]$$

2. Команда $DO[i] = ON/OFF$ включает (ON) или выключает (OFF) заданный цифровой выходной сигнал.

$$DO[i] = ON/OFF$$

3. Команда $DO[i] = PULSE$, [длительность] инвертирует текущее состояние указанного цифрового выхода на заданное время в секундах с дискретностью 0.1С (от 0.1 до 25.5 С).

$$DO[i] = PULSE, [длительность]$$

4. Команда $DO[i] = R[i]$ включает или выключает заданный цифровой выходной сигнал в соответствии со значением заданного регистра. Если заданный регистр имеет значение 0, цифровой выходной сигнал выключается. Если заданный регистр имеет значение, отличное от 0, цифровой выходной сигнал включается.

$$DO[i] = R[i]$$

5.3. КОМАНДЫ АНАЛОГОВОГО ВВОДА-ВЫВОДА СИСТЕМЫ

Аналоговый входной (AI) и аналоговый выходной (AO) сигналы показывают уровни непрерывно изменяющихся значений. Т.е. величина сигнала может представлять аналоговые величины (температуру, напряжение и др.).

1. Команда $R[i] = AI[i]$ сохраняет значение аналогового входного сигнала в регистре.

$$R[i] = AI[i]$$

2. Команда $AO[i] = (\text{значение})$ выводит значение в виде заданного аналогового выходного сигнала.

$$AO[i] = (\text{значение})$$

3. Команда $AO[i] = R[i]$ выводит значение указанного регистра в виде аналогового выходного сигнала.

$$AO[i] = R[i]$$

6. КОМАНДЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ

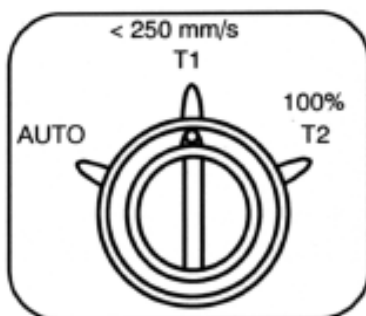
Программа может быть запущена:

- при помощи пульта обучения (клавиша SHIFT и клавиша FWD или BWD) в пошаговом и непрерывном режимах
- с панели оператора, нажав кнопку START (запуск) .

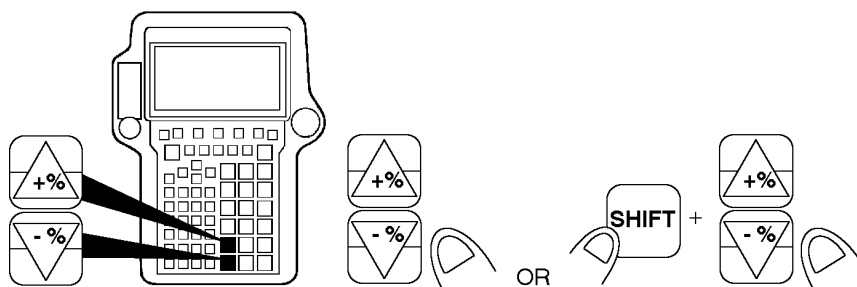
6.1. ЗАПУСК ПРОГРАММЫ ПРИ ПОМОЩИ ПУЛЬТА ОБУЧЕНИЯ В ПОШАГОВОМ РЕЖИМЕ

Выполнение пошагового тестирования (пошаговой работы) - это построчное выполнение программы

1. Установите трехпозиционный переключатель на пульте оператора в положение T1.



2. Включите пульт обучения.
3. Задайте нужную скорость перемещения рабочих органов.



3. Нажмите клавишу SELECT (выбор). Отображается экран выбора программы.
4. Выберите программу, которая будет тестироваться, и нажмите клавишу ENTER. Появится экран редактирования программы.
5. Нажмите клавишу STEP, чтобы выбрать пошаговый режим. Информация STEP на экране пульта обучения загорается желтым цветом.
6. Переместите курсор в строку начала программы.
7. При нажатом переключателе аварийной остановки нажмите RESET. Красный светодиод FAULT на пульте обучения должен погаснуть.
8. Для выполнения программы при нажатой клавише SHIFT нажмите и клавишу FWD (вперед). Не отпускайте клавишу SHIFT, пока выполнение команды не закончено.



8. После выполнения одной строки программы программа приостанавливается.

- При выполнении команды перемещения курсор останавливается в выполненной строке. В следующий раз при запуске выполнения программы в прямом направлении выполняется следующая строка программы.

- При выполнении команды управления курсор перемещается в следующую строку.

9. Нажмите клавишу STEP, чтобы выключить пошаговый режим.

10. Выключите пульт обучения при помощи выключателя, затем отпустите переключатель аварийной блокировки.

6.2. ЗАПУСК ПРОГРАММЫ ПРИ ПОМОЩИ ПУЛЬТА ОБУЧЕНИЯ В НЕПРЕРЫВНОМ РЕЖИМЕ

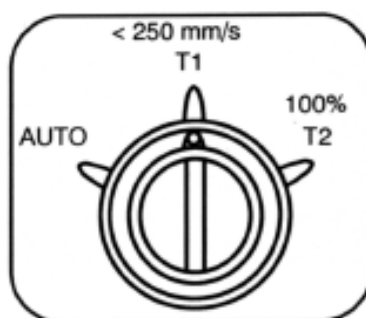
1. Установите трехпозиционный переключатель на пульте оператора в положение T1 или T2.

T1 (<250 мм/с): тестовый режим 1

Скорость как вершины инструмента, так и фланца ограничены и не превышают 250 мм/сек. В режиме T1 невозможно проверить фактическую траекторию инструмента робота и время цикла, потому что рабочая скорость ограничена.

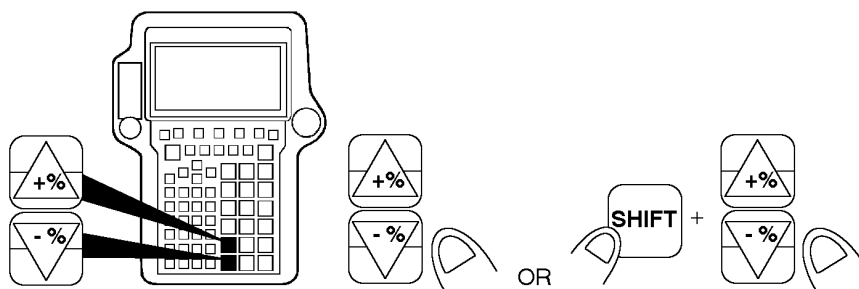
T2 (100%): тестовый режим 2

Режим T2 предназначен для окончательной проверки созданной программы путем прогона робота на скорости заданной в программе, потому что, в принципе, ограничение скорости отсутствует. Однако, при ручном управлении и обучении скорость как вершины инструмента, так и фланца ограничены и не превышают 250 мм/сек.



2. Включите пульт обучения.

3. Задайте нужную скорость перемещения рабочих органов.



4. Нажмите клавишу SELECT (выбор). Отображается экран выбора программы.

5. Выберите программу, которая будет тестироваться, и нажмите клавишу ENTER. Появится экран редактирования программы.

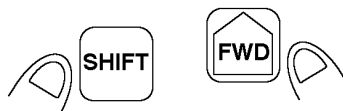
6. Задайте непрерывный режим. Убедитесь, что светодиод STEP горит зеленым цветом.

(Если световой индикатор STEP горит желтым цветом, нажмите клавишу STEP).

7. Переместите курсор в строку начала программы.

8. При нажатом переключателе аварийной остановки нажмите RESET. Красный светодиод FAULT на пульте обучения должен погаснуть.

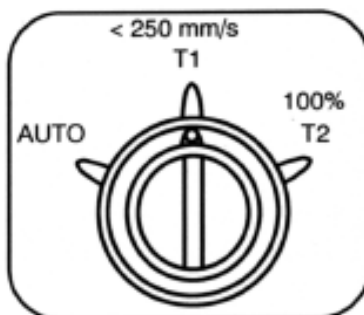
9. Для выполнения программы при нажатой клавише SHIFT нажмите и клавишу FWD (вперед). Не отпускайте клавишу SHIFT, пока выполнение программы не закончено.



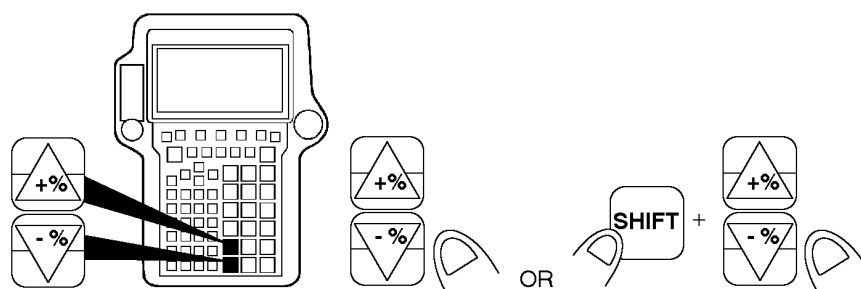
10. Выключите пульт обучения при помощи выключателя, затем отпустите переключатель аварийной блокировки.

6.3. ЗАПУСК ПРОГРАММЫ С ПАНЕЛИ ОПЕРАТОРА

1. Установите трехпозиционный переключатель на пульте оператора в положение AUTO.



2. Задайте нужную скорость перемещения рабочих органов.



3.Нажмите клавишу SELECT (выбор). Отображается экран выбора программы.

4. Выберите программу, которая будет выполняться, и нажмите клавишу ENTER. Появится экран редактирования программы.

5.С помощью клавиши STEP задайте пошаговый или непрерывный режим работы. Если световой индикатор STEP горит желтым цветом – пошаговый режим работы, зеленый – непрерывный. Переключение между режимами с --помощью клавиши STEP).

6. Переместите курсор в строку начала программы.

7. Выключите пульт обучения.

8. Нажмите кнопку CYCLE START на панели оператора. Программа начнет выполняться.



ВНИМАНИЕ!!! Программа будет выполняться только при выключенном пульте обучения. В противном случае на табло пульта обучения загорится ошибка SYST-045. Для ее устранения выключите пульт обучения и нажмите RESET.