



## Краткое руководство по использованию устройства (e)

### 940 Краткое руководство программиста

#### Схема программы:

При разработке программы очень важно правильно проработать ее структуру. Рекомендуется делить программу на 7 секций:

**Header** («Заголовок»): Заголовок определяет название программы, включает имя автора и описание функций программы. Он также может содержать дату и номер.

**I/O List** («Список входов/выходов»): Список входов/выходов указывает, для каких целей используются определенные входы/выходы привода.

**Init & Set Var**: Начальные условия, заданные переменные определяют настройки привода и системные переменные.

**Events** («События»): События представляет собой малую программу, работающую независимо от основной программы. Данный раздел предназначен для указания программных событий.

**Main Pgm** («Основная программа»): Основная программа – это область, в которой определяется работа привода.

**Sub-Routines** («Подпрограммы»): Область, в которой должны находиться все подпрограммы. Эти подпрограммы впоследствии будут вызываться из основной программы посредством команды GOSUB.

**Fault Handler** («Программа устранения неисправностей»): В данной области содержится текст программы – алгоритма устранения неисправностей. Эта часть программы будет срабатывать каждый раз при возникновении неисправностей привода.

#### Включение/отключение

Команда	Синтаксическая структура	Название
DISABLE	DISABLE	Отключение выхода сервопривода
ENABLE	ENABLE	Включение выхода сервопривода

#### Программа обработки событий

Команда	Синтаксическая структура	Название
EVENT	EVENT <имя> INPUT <название входа> EVENT < имя > INPUT !<название входа> EVENT < имя > TIME <интервал времени> EVENT < имя > <expression- выражение>	Запуск программы обработки того или иного события
ENDEVENT	ENDEVENT	Окончание обработки события
EVENT ON/OFF	EVENT < имя > ON EVENT < имя > OFF	Включение/отключение того или иного события
EVENTS ON/OFF	EVENTS OFF EVENTS ON	Отключение всех текущих событий Возобновление обработки всех отключенных событий
JUMP	JUMP <label>	Перейти к названной метке(label) из программы обработки событий

**Операторы, которые нельзя использовать в обработке событий:**  
MOVE, MOVED, MOVEP, MOVEDR, MOVEPR, MDV, MOTION SUSPEND, MOTION RESUME, STOP MOTION, DO UNTIL, GOTO, GOSUB, HALT, VELOCITY ON/OFF, WAIT, WHILE

#### Fault Handling («Программа устранения неисправностей»)

Команда	Синтаксическая структура	Название
ON FAULT	ON FAULT	Запуск программы устранения неисправности
ENDFAULT	ENDFAULT	Остановка программы устранения неисправности
RESUME	RESUME <label>	Выход из программы устранения неисправностей и возобновление работы основной программы через переход на названную метку (label).

**Операторы, которые нельзя использовать в программе устранения неисправности:**  
ENABLE, WAIT UNTIL, MOVE, MOVED, MOVEP, MOVEDR, MOVEPR, MDV, MOTION SUSPEND, MOTION RESUME, GOTO, GOSUB, JUMP, ENABLE, VELOCITY ON/OFF

#### Вход/Выход

Команда	Синтаксическая структура	Название
ASSIGN	ASSIGN INPUT<название входа>AS BIT<bit #>	Назначить вход, как бит №... в INDEX

## Краткое руководство по использованию устройства (е) 940 Краткое руководство программиста

### Разное

Команда	Синтаксическая структура	Название
DEFINE	DEFINE<имя><string-ряд>	Задайте имя состоянию переменной или числу
FAULT	FAULT FaultNumber	Программа пользователя сгенерировала ошибку №...
ICONTROL ON/OFF	ICONTROL ON ICONTROL OFF	Активация интерфейс-управления Деактивация команд с интерфейса
REGISTRATION ON	REGISTRATION ON	Активация регистрационного входа (IN_C3) Запись фактического положения в RPOS
SEND	SEND [ N#]	Передать переменную N0...или N31 другим устройствам (по сети Ethernet)
SEND TO	SENDTO GroupID [N#]	Передать переменную на другой привод (или группу приводов) (по сети Ethernet)

### Команды движения/перемещения

Команда	Синтаксическая структура	Название
MOVE	MOVE [BACK] UNTIL <condition> [ ,C] MOVE [BACK] WHILE <condition> [ ,C]	Двигаться [назад] до выполнения <условия> Двигаться [назад] пока выполняется <условие>
MOVED	MOVED <distance> [ ,S ] [ ,C]	Переместится НА <расстояние> [ S-ускорение]
MOVEP	MOVEP <absolute position> [ ,S ] [ ,C]	В <позицию> [C-двигаясь, наблюдать за входами]
MOVEDR	MOVEDR <distance> , <displacement> [ ,C]	Движение НА... с регистрац. изменения на входе
MOVEPR	MOVEPR <distance> , <displacement> [ ,C]	Движение В... с регистрац. изменения на входе
MDV	MDV <[-]segment distance> , <segment final velocity> [ ,S]	Сегментное движение-расстояние> , <скорость в конце сегмента>
MOTION SUSPEND	MOTION SUSPEND	Временная приостановка движения
MOTION RESUME	MOTION RESUME	Возобновление приостановленного движения
VELOCITY MODE ON/OFF	VELOCITY ON VELOCITY OFF	Включение/выключение режима скорости

### Структура программы

Команда	Синтаксическая структура	Название
DO / UNTIL	DO<statement(s)> UNTIL<condition>	Выполнять <оператор(ы)> пока выполняется <условие>
END	END	Завершение программы
GOTO	GOTO <label>	Перейти на текстовую метку
GOSUB	GOSUB<название подпрограммы>	Перейти к подпрограмме<название подпрограммы>
HALT	HALT	Приостановить работу программы
IF / ELSE/ ENDIF	IF <condition> ELSE ENDIF	Условный переход - Если / В противном случае / Завершить выполнения сравнения
STOP MOTION STOP MOTION QUICK	STOP MOTION STOP MOTION QUICK	Остановить все движения с замедлением DECEL Остановить все движения с замедлением QDECEL
WAIT	WAIT UNTIL <expression> WAIT WHILE <expression> WAIT TIME <time delay> WAIT MOTION COMPLETE	Ждать до выполнения<выражение> Ждать пока выполняется <выражение> Ждать время Ждать завершения движения
WHILE / ENDWHILE	WHILE <expression> ENDWHILE	Выполнять пока справедливо <выражение> Конец выполнения

