

TITANIUM[®]



Часть II

Конфигурирование и язык электроавтоматики

Оглавление

II	Конфигурирование и язык электроавтоматики	1
1	Описание языка электроавтоматики	6
1.1	Структура программы	6
1.2	Элементы языка	9
1.2.1	Заголовок программы	10
1.3	Директивы	11
1.3.1	Директива инициализации переменных	11
1.3.2	Специальные директивы	12
1.3.3	Быстрая и медленная секции	12
1.4	Уравнения	14
1.5	Переменные	15
1.6	Константы	19
1.7	Операторы	19
1.7.1	Логические операции	19
1.7.2	Арифметические операции	21
1.7.3	Условные операторы	22
1.7.4	Безусловный переход	23
1.8	Промежуточные ячейки динамической памяти (M)	24
1.9	Промежуточные ячейки статической памяти (D)	25
1.10	Ячейки статической памяти управляющих программ (R)	26
1.11	Системные параметры пользователя	26
1.12	Обменные ячейки	27
1.12.1	Общий перечень обменных ячеек	28
1.13	Таймеры	47
1.13.1	Таймер задержки установки	48
1.13.2	Таймер задержки сброса	49
1.14	Счетчики	51
1.15	Одновибраторы	52



1.16	Специальные функции и переменные	53
1.16.1	Функция «выхода в 0» по оси	54
1.16.2	Функция прекращения «выхода в 0»	55
1.16.3	Формирование сообщений и ошибок	55
1.16.4	Работа с программируемой виртуальной клавиатурой	56
1.16.5	Функция поиска в таблице инструментов	57
1.16.6	Функция управления осью от ПЛК	57
1.16.7	Функция привязки текущего инструмента от ПЛК	59
1.16.8	Функция задания смещения от ПЛК	59
1.16.9	Функция задания коррекции смещения от ПЛК	59
1.16.10	Функция увеличения коррекции смещения от ПЛК	60
1.16.11	Функция задания высоты текущего инструмента от ПЛК	60
1.16.12	Функция задания высоты корректора от ПЛК	60
1.16.13	Функция получения номера слота	61
1.16.14	Функция получения номера инструмента в слоте	61
1.16.15	Функция вставки инструмента в слот	61
1.16.16	Функция получения количества магазинов инструментов	62
1.16.17	Функция получения кода ошибки магазина инструментов	62
1.16.18	Функция получения типа магазина инструментов	62
1.16.19	Функция получения количества слотов в магазине инструментов	63
1.16.20	Функция получения номера слота в магазине инструментов по его идентификатору	63
1.16.21	Функция получения идентификатора слота в магазине инструментов по его номеру	63
1.16.22	Функция получения идентификатора инструмента по номеру слота в магазине инструментов	64
1.16.23	Функция получения координат инструмента по номеру слота в магазине инструментов	64
1.16.24	Функции получения/записи битов переменной	64
1.16.25	Функции режима обучения	65
1.16.26	Функции создания и работы с пультом быстрых программ	67
2	Система параметров устройства ЧПУ	70
2.1	Системные параметры	71
2.2	Общие базовые станочные параметры	87
2.3	Общие станочные параметры	88
2.4	Общие технологические параметры	89
2.5	Общие наладочные параметры	94
2.6	Базовые станочные параметры оси	94
2.7	Станочные параметры оси	97
2.8	Наладочные параметры оси	101



2.9	Параметры пользователя	103
2.10	Базовые станочные параметры маховика	103
2.11	Настройка модулей	104
2.11.1	Настройка драйвера блока ЦАП	105
2.11.2	Настройка драйвера блока оцифровки	105
2.11.3	Настройка входов	106
2.11.4	Настройка выходов	106
2.12	Настройка параметров для пульта оператора	107
2.12.1	Настройка пульта оператора spc11	107
2.12.2	Настройка пульта оператора НВ-04	108
2.13	Перечень параметров, необходимых для организации управления приводами подач	110
2.14	Настройка осей	111
2.14.1	Дискретность оси для круговых датчиков	111
2.14.2	Настройка выхода в 0	112
2.14.3	Настройка безопасного подвода осей	114
2.14.4	Настройка подготовки исполнения со строки	114
3	Руководство по сопряжению устройства ЧПУ со станками.	115
3.1	Создание нового файла электроавтоматики	116
3.2	Включение станка	121
3.3	Включение приводов	122
3.4	Сигналы подготовки перемещения	123
3.5	Выбор режима работы	124
3.6	Настройка приводов	126
3.6.1	Алгоритм базовой настройки привода	126
3.7	Пуск управляющей программы	127
3.8	Ручной режим	128
3.9	Смена инструмента	128
3.10	Управление шпинделем	129
3.10.1	Включение шпинделя	129
3.10.2	Реверс шпинделя	131
3.10.3	Конфигурации с несколькими шпинделями	133
3.10.4	Подчинённые оси	133
4	Диагностика проблем	135
4.1	Диагностика датчиков обратной связи	135
4.1.1	Неисправность инкрементального датчика	135
4.2	Диагностика движения по осям	136
4.2.1	Неверные параметры приводов	136

Введение

Язык электроавтоматики (ЭА) системы ЧПУ TITANIUM® (далее CNC-11) предназначен для разработки программного модуля привязки электрооборудования станка к системе. Язык ЭА представляет собой адаптированную разновидность языка Си. Программа электроавтоматики выполняется параллельно с программным модулем системы. Целью работы программы ЭА является управление оборудованием станка и согласование его работы с работой программного обеспечения системы ЧПУ.

ВНИМАНИЕ!

После отладки программы электроавтоматики, установки параметров и наладки оборудования всю ответственность за любые изменения программы и параметров несет тот, кто выполнил эти изменения!

Описание языка электроавтоматики

1.1 Структура программы

Программа ЭА состоит из шести частей, которые объединены в одном файле. Первая часть называется **заголовком** и является общей для остальных пяти частей.

```
1 #define PLC_PROGRAM
2 #include "XXXXX.h"
3 PLC_TITLE_START
4 #define SB_AUTO_MODE I203_1
5 // ...
6 PLC_TITLE_END
```

ВАЖНО: заголовок всегда должен начинаться со служебных строк:

```
1 #define PLC_PROGRAM
2 #include "XXXXX.h"
3 PLC_TITLE_START
```

и заканчиваться строкой:

```
1 PLC_TITLE_END
```

где XXXXX — имя программы.

Пример. Заголовок программы



```
1 #define PLC_PROGRAM
2 #include <cnc-plc/template/plc_user.h>
3 PLC_TITLE_START
4 // ...
5 // Объявление переменных
6 #define SB_AUTO_MODE I203_1
7 // ...
8 PLC_TITLE_END
```

В заголовке объявляются константы, которые будут использоваться в исполняемой части программы, а также псевдонимы (символические имена) переменных.

Пример. Объявление констант и переменных (присвоение им символьных имен)

```
1 #define SB_RESET      I207_5
2 #define HL_RESET      U237_7
3 #define Error_Kont_Obr M2_1
4 #define Error_Ograzhd M2_2
5 // Постоянные установки
6 #define MaxKod_S_per_1 900
7 #define MaxKod_S_per_2 2000
```

Инициализация переменных.

В этой части программы присваиваются начальные значения счетчиков, таймеров, входов и выходов.

Пример. Инициализация переменных

```
1 PLC_VAR_INIT_START
2 // ...
3 // Инициализация счетчиков, таймеров и переменных
4 T15 = P1 * 10;
5 CV1 = 20;
6 CV2 = 40;
7 T_spind = D1;
8 // ...
9 PLC_VAR_INIT_END
```



Следующие две части программы выполняются **до** и **после перезагрузки** программы ПЛК. Их использование **не обязательно**.

Пример. Секции, выполняющиеся до и после перезагрузки ПЛК-модуля

```
1 PLC_BEFORE_RELOAD_START
2 // Код, исполняемый до перезагрузки
3 PLC_BEFORE_RELOAD_END
4
5 PLC_AFTER_RELOAD_START
6 // ПОСЛЕ
7 PLC_AFTER_RELOAD_END
```

Затем идет часть, выполняемая при завершении работы программы ПЛК. Она также является дополнительной и **не обязательной** в использовании.

Пример. Часть завершения работы ПЛК-модуля

```
1 PLC_FREE_START
2 // ...
3 PLC_FREE_END
```

Последние секции программы называются **быстрой** и **медленной**. Предназначены для описания исполняемого кода и отличаются только частотой запуска. Частота запуска секции задается системными параметрами N109, N110 и определяется разработчиком программы ЭА, исходя из требований к работе электрооборудования.

Пример. Быстрая и медленная секции программы

```
1 PLC_FAST_START
2 // ...
3 // БЫСТРАЯ СЕКЦИЯ
4 AX1_LIM_PLUS_0 = !xSQ_0gr_plus_ && !SB_DRIVES_ON;
5 AX4_LIM_MINUS_0 = 0;
6 AX2_LIM_HOME_0 = !SQ_Zero_Y;
7 ERR(!MACH_NO_ERR_0, 94, "АВАРИЙНЫЙ ОСТАНОВ ПРОГРАММЫ");
8 // ...
9 PLC_FAST_END
```



```
10
11 PLC_SLOW_START
12 // ...
13 // МЕДЛЕНАЯ СЕКЦИЯ
14 KV_kr = TOS1 && (Error_Driver || Error_Kont_Obr || !Stanok_On)
      || !MACH_NO_ERR_0;
15 KV_PP = TOS24;
16 Driver_On = (SB_DRIVES_ON || Driver_On)
17             && !(SB_DRIVES_OFF && !CH1_PRG_RUNNIG_I)
18             && MACH_UI_RDY_I && MACH_NO_ERR_0
19             && Stanok_On
20             && !Error_Driver
21             && !V61_7;
22 // ...
23 PLC_SLOW_END
```

ВАЖНО: в быструю секцию следует помещать только те уравнения, которые нуждаются в быстрой обработке.

Принцип работы секций состоит в следующем: при прохождении заданного интервала времени полностью выполняется заданный код по порядку записи с начала и до конца, причем время выполнения одного цикла обсе-та всегда меньше этого интервала. Порядок записи секций произвольный. Секция состоит из уравнений и комментариев. Уравнения отделяются друг от друга символом «точка с запятой». Уравнения могут располагаться на нескольких строках, а также в одной строке несколько уравнений. Допускается также в строки с уравнениями для наглядности включать произвольное количество пробелов и табуляций. При этом нужно иметь в виду, что не допускается установка пробелов и табуляций в середине имен переменных.

1.2 Элементы языка

Элементами языка являются директивы; переменные («I» — вход, «U» — выход, «M» — промежуточная ячейка памяти, «V» — обменная ячейка, «T» — таймер, «C» — счетчик, «D» — статическая память, «P» — системные параметры пользователя, «R» — одновибраторы (генераторы одиночных импульсов)); функциональные инструкции; числовые константы; логические операторы; арифметические операторы; команды; метки; комментарии.

Допустимые диапазоны задания элементов языка:

- входы I — 1...1023 байта;
- выходы U — 1...1023 байта;



- промежуточная динамическая память (M) — 1...4095 байтов;
- таймеры T — 1...255;
- счетчики C — 1...255;
- статическая память D — 1...255 байтов;
- параметры пользователя P — 1...32 двойных слова (для представления целых чисел);и (PU) — 1...16 значений, для действительных чисел с двойной точностью (8 байт);
- одновибраторы (R), 1...255;
- количество функциональных инструкций — 7;
- максимальное количество меток в программе — неограниченно;
- максимальное количество псевдонимов в программе — неограниченно;
- максимальная длина псевдонима — 20;
- максимальное количество вложений скобок в одном уравнении — неограниченно.

1.2.1 Заголовок программы

В этой части задаются символьные имена переменных и констант.

Пример. Заголовок программы

```
1 #define PLC_PROGRAM
2 #include <cnc-plc/template/plc_user.h>
3 PLC_TITLE_START
4 #define SB_AUTO_MODE      I203_1
5 #define HL_AUTO_MODE      U201_7
6 #define Error_peredacha  M1_3
7 #define kod                M36_W
8 #define peredacha          M56_B
9 #define SF_Stankok_On      !I1_2
10 #define Key_ESC            MACH_UI_ESC_IO
11 #define Kod_S_per_1        900
12 #define Kod_S_per_2        2000
13 PLC_TITLE_END
```



Объявление переменной/константы начинается со служебного слова **#define**, далее указывается имя, а затем — значение. Допускается произвольное количество пробелов/табуляций при объявлении.

ВАЖНО: имена переменных — регистрозависимы (*ERROR*, *Error* и *error* — разные переменные).

ВАЖНО: символ ';' (точка с запятой) не ставится после объявления переменных/констант!

ВАЖНО: не допускается объявление нескольких переменных/констант на одной строке!

1.3 Директивы

Директивы — это указания для компилятора программы электроавтоматики, и они не являются частью исполняемого кода. Появление директивы в тексте программы отменяет действие предыдущей.

ВАЖНО: объявление директив заголовка, инициализации, быстрой и медленной секций **ОБЯЗАТЕЛЬНО** вне зависимости от необходимости!

1.3.1 Директива инициализации переменных

В этой части задаются начальные значения таймеров, счетчиков, входов/выходов и переменных. При необходимости также допускается использование уравнений и функций в этой части программы.

Пример. Инициализация переменных

```
1 PLC_VAR_INIT_START
2 CH1_MAN_0 = 1;    // Включить ручной режим.
3 // ТАЙМЕРЫ
4 T1 = 5; T2 = 5;  // Генератор на индикацию.
5 T3 = 200;        // Включение привода S на слежение.
6 T4 = 10;         // Задержка на поиск ноль метки при M19.
7 T15 = P1 * 10;  // Время смазки.
8 // СЧЁТЧИКИ
9 CV1 = 20;        // Счетчик левого магазина.
10 CV2 = 40;       // Счетчик правого магазина.
11 T_spind = D1;   // Восстановить инструмент в шпинделе.
12 First_step = 1;
13 podacha = 1;
14 // ...
15 PLC_VAR_INIT_END
```



ВАЖНО: во всех частях программы, **кроме заголовка**, допускается написание нескольких уравнений и функций или на одной строке, разделенных символом ';' (точка с запятой).

1.3.2 Специальные директивы

Следующие две части программы выполняются **до** и **после перезагрузки** программы ПЛК. В них могут выполняться любые логические или математические операции.

Пример. Директивы, выполняющиеся до и после перезагрузки ПЛК-модуля

```
1 PLC_BEFORE_RELOAD_START
2 // ДО
3 PLC_BEFORE_RELOAD_END
4
5 PLC_AFTER_RELOAD_START
6 // ПОСЛЕ
7 PLC_AFTER_RELOAD_END
```

Затем идет часть, выполняемая при завершении работы программы ПЛК.

Пример. Директива завершения работы ПЛК-модуля

```
1 PLC_FREE_START
2 // ...
3 PLC_FREE_END
```

ВАЖНО: специальные директивы применяются крайне редко и их использование **не обязательно!**

1.3.3 Быстрая и медленная секции

Последние секции программы называются быстрой и медленной. Они предназначены для описания исполняемого кода и отличаются только частотой запуска. Частота запуска сверхбыстрой, быстрой и медленной секций задается системными параметрами N108, N109, N110 соответственно и определяется разработчиком программы ЭА, исходя из требований к работе электрооборудования.

Периоды запуска быстрой (N109) и медленной (N110) секций должны быть кратными периоду запуска сверхбыстрой секции (N108), иначе берётся наименьшее кратное значение.

**Пример.** Расчет времени работы секций
$$N_{108} = 2 \text{ (частота запуска сверхбыстрой секции)}$$
$$F_{\text{fast}} = 2 * 5 = 10 \text{ мс}$$
$$F_{\text{slow}} = 2 * 50 = 100 \text{ мс (т. е. через каждые 100 итераций}$$

быстрой секции один раз будет выполняться медленная)

Директива PLC_FAST_START сообщает транслятору о том, что код программы, приведенный ниже, следует разместить в быстрой (высокочастотной секции).

Пример. Быстрая секция

```
1 PLC_FAST_START
2 // ...
3 // Аварийный останов
4 MACH_NO_ERR_0 = MACH_UI_RDY_I && !Error_Stanok_On && !
   Error_Driver && !Error_peredacha && !Error_sm_pered& !
   Error_sm_T && !Error_Shpind;
5 ERR(!MACH_NO_ERR_0, 94, "АВАРИЙНЫЙ ОСТАНОВ ПРОГРАММЫ");
6 // ...
7 PLC_FAST_END
```

Необходимо помнить, что в быструю секцию следует помещать только те уравнения, которые нуждаются в быстрой обработке (например, контроль ограничителей или аварийных ситуаций).

Принцип работы секции состоит в следующем: при прохождении заданного интервала времени полностью выполняется заданный код по порядку записи с начала и до конца, причем время выполнения одного цикла обсчета всегда меньше этого интервала.

ВАЖНО: Порядок записи секций произвольный.

Медленная и быстрая секции состоят из уравнений, функций, операторов и комментариев.

ВАЖНО: Уравнения отделяются друг от друга символом ';' ("точка с запятой"). Уравнения могут располагаться на нескольких строках, а также в одной строке несколько уравнений. Допускается также в строки с уравнениями для наглядности включать произвольное количество пробелов и табуляций. При этом нужно иметь в виду, что не допускается установка пробелов и табуляций в середине имен переменных (например, между символом переменной и ее номером, или между номером байта и номером бита, и т. д.)!



Директива начала медленной секции (PLC_SLOW_START) сообщает транслятору о том, что код программы, приведенный ниже, следует разместить в медленной (низкочастотной секции).

Пример. Медленная секция программы

```
1 PLC_SLOW_START
2 // ...
3 // ВКЛЮЧЕНИЕ СТАНКА
4 Stanok_On = (SB_STANOK_ON || (Stanok_On && SF_Stanok_On))
5           && !(SB_STANOK_OFF && !CH1_PRG_RUNNIG_I)
6           && MACH_NO_ERR_0
7           && MACH_UI_RDY_I
8           && !Error_Stanok_On
9           && (i_No_SQ_Avar_XYZ || SB_STANOK_ON)
10          && !TOS29;
11 // ...
12 PLC_SLOW_END
```

Директива PLC_SLOW_END сообщает транслятору о завершении медленной секции.

Большая часть логики работы станка описывается в медленной секции: включение станка/приводов, выбор режимов работы, управление в ручном и автоматическом режимах, обработка М-кодов и т. д.

1.4 Уравнения

Исполняемые части программы включают в себя уравнения. Уравнение состоит из правой и левой части и заканчивается символом «;». В левой части указывается переменная, счетчик, таймер или одновибратор. В правой части вычисляется значение, которое будет передано в левую часть уравнения.

Левая и правая часть простого уравнения разделяются символом «=» (присвоить).

В уравнениях, которые используются для проверки условий, применяются следующие операторы: «==» (равно), «<=» (меньше либо равно), «>=» (больше либо равно), «!=» (не равно), «>» (строго больше), «<» (строго меньше).

Пример. Уравнения



```
1 // ...
2 U1_4 = I1_1
3     || M(2_3 || (T0S1 || !I3_2));
4 U3_W = 1204; U2_B = 40;
5 I2_4 = (C52 >= 4);
6 R23 = !I1_2 && U1_3;
7 I5_B = (T0S2 >= 6) * 4;
8 // ...
```

В первом уравнении для наглядности элементы правой части разбиты на несколько строк с несколькими пробелами и табуляциями между переменными. На строке ниже несколько уравнений, разделенные символом ';' («точка с запятой»), расположены на одной строке.

ВАЖНО: Уравнения отделяются друг от друга символом ';' ("точка с запятой"). Уравнения могут располагаться на нескольких строках, а также в одной строке несколько уравнений. Допускается также в строки с уравнениями для наглядности включать произвольное количество пробелов и табуляций. При этом нужно иметь в виду, что не допускается установка пробелов и табуляций в середине имен переменных (например, между символом переменной и ее номером, или между номером байта и номером бита, и т. д.)!

1.5 Переменные

Переменные программы ЭА могут быть 4 типов: бит, байт, слово и двойное слово. Переменные делятся на несколько групп:

- входные сигналы (обозначаются символом «I»);
- выходные сигналы (обозначаются символом «U»);
- переменные, которые можно использовать для собственных нужд ПЛК (обозначаются символом «M»);
- ячейки постоянной памяти, которые можно использовать для собственных нужд ПЛК (обозначаются символом «D»);
- ячейки постоянной памяти, которые можно использовать для связи управляющих программ и ПЛК (обозначаются символом «GR»);
- обменные ячейки памяти, отвечающие за взаимодействие ЧПУ и модуля ПЛК (обозначаются символом «V»);



- одновибраторы (обозначаются символом «R»);
- таймеры (обозначения: «T», «TS», «TR», «TOR», «TOS»);
- счетчики (обозначаются символом «C»);
- системные параметры пользователя (обозначения: «P», «PU»).

Каждая группа переменных располагается в памяти ЧПУ в виде последовательности байт. Объем памяти, выделяемый под каждую группу, определяется автоматически в процессе трансляции программы и имеет фиксированный размер.

Порядковый номер переменной, которым является номер байта относительно начала указанной области памяти, определяется числом, следующим за символом переменной (справедливо для I, U, M, D, GR, V). Нумерация начинается с 1. Тип переменной указывается после символа « » (нижнее подчеркивание), который должен быть записан после номера переменной. Числам от 1 до 8 соответствует номер бита в байте, номер которого указан до нижней черты. Таким образом определяется битовая переменная. Символ «B» определяет переменную типа байт, символ «W» — слово (2 байта), символ «D» — двойное слово (4 байта).

В таблице 1.5.1 представлена адресация переменных на основе M-ячеек. Данная структура справедлива также для входов (I), выходов (U) и ячеек статической памяти (D).

В обменных ячейках (V) после « » (нижнее подчеркивание) также указан размер ячейки, однако он не подчиняется общему правилу, а лишь указывает на размер памяти, выделенный под ячейку, и структурирует данные.

После переменной T указывается тип таймера: TS — таймер с задержкой на установку, TR — таймер с задержкой на сброс. Далее в скобках указан входной сигнал этого таймера.

Для счетчиков после переменной C через « » (нижнее подчеркивание) указывается тип входа: символ «I» — прибавляющий вход счетчика или символ «D» — вычитающий вход.

Одновибраторы обозначаются символом R и работают только с битовыми значениями. Вход вибратора указывается в левой части уравнения, обозначается через RI.

Параметры пользователя хранятся в файле конфигурации. Они разделены на две группы: целочисленные P и действительные PU.

Пример. Описание переменных



Таблица 1.5.1: Структура адресации ячеек

Адрес (бит)	Адрес (байт)	Адрес (слово)	Адрес (двойное слово)
M1_1	M1_B	M1_W	M1_D
M1_2			
M1_3			
M1_4			
M1_5			
M1_6			
M1_7			
M1_8			
M2_1	M2_B	M2_W	
M2_2			
M2_3			
M2_4			
M2_5			
M2_6			
M2_7			
M2_8			
M3_1	M3_B	M2_W	
M3_2			
M3_3			
M3_4			
M3_5			
M3_6			
M3_7			
M3_8			
M4_1	M4_B		
M4_2			
M4_3			
M4_4			
M4_5			
M4_6			
M4_7			
M4_8			



```
1 I1_2    // Второй бит первого байта I-ячейки
2 U2_B    // Вторая U-ячейка как байт
3 GR4_B   // Четвертый байт R-ячейки
4 M200_W  // Два байта M-ячейки (200-201 байты)
5 M220_D  // Четыре байта M-ячейки (220-223 байты)
6 V34_8   // Восьмой бит 34-ой обменной ячейки (обращение как к
           // байту НЕВОЗМОЖНО!)
7 V33_W   // 33-я обменная ячейка, содержащая числовое значение
           // вещественного или целого типа
8 TS4()   // Задание входного значения для четвёртого таймера
9 RI2     // Вход второго одновибратора
10 PU5    // Пятый действительный параметр пользователя
11 CD1()  // Вычитающий вход первого счетчика
```

Для упрощения работы с переменными и константами существует возможность обращения к ним через символьные имена, объявленные в заголовке программы.

Пример. Присвоение символьных имен переменным и константам

```
1 PLC_TITLE_START
2 #define SB_STANOK_ON      I203_5
3 #define HL_STANOK_ON      U205_7
4 #define SB_STANOK_OFF     I203_6
5 #define HL_STANOK_OFF     U206_7
6 #define Error_Stanok_On   M1_1
7 #define Stanok_On         M45_3
8 #define SF_Stanok_On      I1_1
9 #define KV_Stanok_On      U1_1
10 #define MaxKod_S_per_2_G96 300
11 #define Kod_S_per_1       100
12 PLC_TITLE_END
```

ВАЖНО: при описании переменных в заголовке использование пробелов, точек, табуляций ЗАПРЕЩЕНО. Также запрещено использование имен, начинающихся с цифры.

Для описания кнопок пульта рекомендуется использовать символьные имена формата:

SB_<имя_кнопки> – для кнопок

HL_<имя_кнопки> – для диода



ВАЖНО: ДЛЯ ПУЛЬТА ОПЕРАТОРА: нумерация диодов (U — выходов) начинается с 1-го байта, номер которого указан в параметре N116, причем используется только 7-й бит каждого байта. Нумерация кнопок (I — входов) начинается с 1-го байта, номер которого указан в параметре N115. Первые два байта — энкодеры. Схема нумерации клавиш пульта оператора см. в «Руководстве по эксплуатации».

1.6 Константы

В правой части уравнения допускается использовать числовые константы. Константы могут задаваться в десятичной (по умолчанию), шестнадцатеричной (с использованием символов **0x**) и восьмеричной системах счисления (с использованием символов **0**).

Пример. Использование разных типов констант

```
1 U2_B = 0x06cd; // 0x06cd – шестнадцатеричная константа
2 M2_W = 243;    // 243 – десятичная константа
3 I2(077);      // 077 – восьмеричная константа
```

Пример. Объявление констант в заголовке

```
1 #define MaxKod_S_per      900
2 #define Kod_S_per         1200
```

1.7 Операторы

Для реализации логики работы программы электроавтоматики используются операторы. Различают логические, арифметические и специальные операторы (условия и переходов). Также в языке электроавтоматики можно использовать любые операторы и типы языка Си (стандарт C11 с использованием расширений GNU, именуемый как GNU11).

1.7.1 Логические операции

Логические операции выполняются над битовыми переменными и числовыми константами, которые указаны в правой части уравнения. Приоритеты операций (в порядке понижения) следующие:



- логическое отрицание (операция НЕ, "!");
- логическое умножение (операция И, "&&");
- логическое сложение (операция ИЛИ, "||").

Последовательность вычислений определяется приоритетом операций, то есть первой будет выполняться операция с высшим приоритетом.

Пример. Последовательность логических операций

```
1 U1_1 = I2_3 || M2_1 && !I1_6;
```

Последовательность вычислений будет следующей:

- !I1.6 — логическое отрицание (операция НЕ с сигналом I1_6);
- M2_1 && !I1.6 — логическое умножение (операция И сигнала M2_1 и результата первой операции);
- I2_3 || M2_1 && !I1.6 — логическое сложение (операция ИЛИ сигнала I2_3 с результатом второй операции).

Последовательность вычислений можно изменить с помощью скобок.

Пример. Изменение последовательности логических операций

```
1 U1_1 = (I2_3 || M2_1) && !I1_6;
```

Последовательность вычислений будет следующей:

- I2_3 || M2_1 — логическое сложение (операция ИЛИ сигнала I2_3 с сигналом M2_1);
- !I1_6 — логическое отрицание (операция НЕ с сигналом I1_6);
- (I2_3 || M2_1) && !I1_6 — логическое умножение (операция И результата первой операции с результатом второй операции).

Вычисления происходят по "короткой схеме", то есть если результат не изменится в процессе дальнейших операций, то вычисления заканчиваются.

Пример. Короткая схема вычислений



```
1 U1_1 = I2_3 || M2_1 && !I1_6;  
2 // Если сигнал I2_3 равен 1, то остальная часть уравнения не  
   // вычисляется и сигнал U1_1 устанавливается в единичное  
   // состояние (включается).  
3 U1_1 = I2_3 && M(2_1 || !I1_6);  
4 // Если сигнал I2_3 равен 0, то остальная часть уравнения не  
   // вычисляется и сигнал U1_1 устанавливается в нулевое состояние  
   // (выключается).
```

При составлении уравнений следует помнить о «короткой схеме» и указывать наиболее быстрый порядок вычислений.

Пример. Короткая схема вычислений

```
1 U1_1 = I2_3 && M(2_1 || !I1_6);  
2 U1_1 = (M2_1 || !I1_6) && I2_3;
```

В случае нулевого сигнала I2_3 первое уравнение будет выполнено быстрее.

1.7.2 Арифметические операции

Арифметические операции выполняются над переменными и числовыми константами, которые указаны в правой части уравнения. Приоритеты операций (в порядке понижения) следующие:

- умножение и деление «*», «/»;
- сложение и вычитание «+», «-».

Последовательность вычислений определяется приоритетом операций, то есть первой будет выполняться операция с высшим приоритетом.

```
1 U1_W = I2_ — M2_W * I1_;
```

Последовательность вычислений будет следующей:

- M2_W * I1_ — умножение значения M2_W на значение I1_;
- I2_ — M2_W * I1_ — вычитание результата первой операции из I2_.

Последовательность вычислений можно изменить с помощью скобок.



Пример. Арифметическая операция

```
1 U1_W = (I2_V — M2_W) * I1_V;
```

Последовательность вычислений будет следующей:

- $I2_V - M2_W$ — вычитание значения $M2_W$ из $I2_V$;
- $(I2_V - M2_W) * I1_V$ — умножение результата первой операции на $I1_V$.

1.7.3 Условные операторы

Оператор ветвления (условная инструкция, условный оператор) — оператор, обеспечивающий выполнение определенной команды (набора команд) только при условии истинности некоторого логического выражения, либо выполнение одной из нескольких команд (наборов команд) в зависимости от значения некоторого выражения.

Синтаксис оператора следующий:

```
1 if условие(<1>) {операторы
2 <1>
3 }
4 else {операторы
5 <2>
6 }
```

Если условие истинно (равно 1), то выполняется блок операторов 1, в противном случае выполняется блок 2. В зависимости от реализуемой логики оператора **else** может и не быть. Так же допускается вложенность операторов (в качестве операторов могут быть использованы условные операторы).

Пример. Вложенность операторов

```
1 if условие(<1>) {операторы
2 <1>
3 if условие(<2>) {операторы
4 <1_1>
5 }
6 } else {
```



```
7  if условие(<2>) {операторы
8  <2>
9  } else {операторы
10 <2_1>
11 }
12 }
```

В примере выше блок «<операторы1>» выполнится в случае, если условие «<условие1>» — истинно, далее будет произведена проверка условия «<условие2>» и в случае его истинности будет выполнен блок «<операторы1_1>». В случае, если условие «<условие1>» ложно, то алгоритм продолжится по ветке else, в которой содержится другой оператор условия.

Блоки операторов **всегда** выделяются фигурными скобками.

Пример. Использование условного оператора

```
1  if (CH1_M_I) {
2  if (CH1_M_R == 2) {
3  CH1_PRG_FIN_O = true;
4  CH1_M_O = true;
5  }
6  }
```

Если пришел сигнал М-команды (CH1_M_I) и ее код равен 2, то даем ответ интерполятору (CH1_M_O) о необходимости завершения управляющей программы (CH1_PRG_FIN_O). При проверке ячейки CH1_M_I не использовалось явное сравнение с true (или 1), так как любое число, не равное нулю, будет интерпретироваться условным оператором как «истина».

1.7.4 Безусловный переход

В программе допускается использование команды безусловного перехода. Для этого используется понятие меток.

Безусловный оператор **goto** позволяет осуществить передачу управления на любую часть программы, имеющую метку, но только в пределах одной секции.

Оператор **goto** записывается в виде:

```
1  goto метка<>;
```



После оператора **goto** безусловно осуществляется переход на тот участок программы, метка которого совпадает с меткой «<метка>».

ВАЖНО: *рекомендуется по возможности избегать использование оператора **goto**, так как его применение делает код программы трудно воспринимаемым!*

При использовании команд условных и безусловных переходов нужно учитывать следующие **правила**:

- нумерация меток производится независимо для каждой секции, поэтому допускается использовать одинаковые номера в разных секциях;
- безусловный переход может быть выполнен как вниз, так и вверх по тексту программы;
- переход допускается только внутри секции, переход в другую секцию запрещен.

Пример. Безусловный оператор

```
1 GOTO L1;
2 U1_1 = I2_3 && !I1_6; (1)
3 U2_1 = I5_2 || M2_1; (2)
4 L1:
5 U3_1 = (I4_2 || M2_4) && !I1_6; (3)
```

В примере будет выполнено только уравнение (3), так как на него будет выполнен переход по метке L1, минуя уравнения (1) и (2).

1.8 Промежуточные ячейки динамической памяти (М)

Промежуточные ячейки динамической памяти предназначены для хранения в них промежуточных результатов работы программы электроавтоматики. Обращение к ним осуществляется через символ «М». Адресация аналогична переменным входных и выходных сигналов. Динамическая память доступна как для чтения, так и для записи. Распределение этих ячеек по назначению произвольное и определяется разработчиком программы электроавтоматики. При выключении устройства ЧПУ информация в них теряется, при включении эти ячейки обнуляются.

Существует возможность редактировать содержимое ячеек в процессе работы станка. Необходимо перейти на вкладку «Ячейки памяти» в меню



«Диагностика» (либо использовать комбинацию клавиш Alt+M для быстрого перехода), выбрав соответствующий пункт в правом меню.

1.9 Промежуточные ячейки статической памяти (D)

Свойства промежуточных ячеек статической памяти аналогичны динамической памяти за исключением того, что информация в них сохраняется при выключении устройства ЧПУ и восстанавливается при его включении. Обращение к ним осуществляется через символ «D». Кроме того, статическая память доступна для редактирования в соответствующих режимах устройства ЧПУ. Поэтому она может использоваться для сохранения текущего состояния станка перед его выключением и для установки этого состояния в режиме редактирования. К текущему состоянию станка можно отнести номер инструмента в шпинделе, соответствие номеров инструмента и номеров гнезд в магазине инструментов и т.д.

Описание назначения статической памяти должно быть указано в документации на конкретный станок. Существует возможность редактировать содержимое ячеек статической памяти в процессе работы станка. Необходимо перейти на вкладку «Ячейки памяти» в меню «Диагностика» (либо использовать комбинацию клавиш Alt+D для быстрого перехода). После этого на дисплее появится окно редактора, в котором будет доступно несколько режимов для редактирования соответствующей ячейки. Возможна запись байта, слова, либо отдельных битов байта. По умолчанию все D-ячейки беззнаковые. Примеры обращений к D-ячейкам приведён в таблице 1.9.1.

Для удобства можно обращения к D-ячейкам обворачивать в макросы.

Пример. Макросы для доступа к D-ячейкам

```
1 // Байт D1
2 #define PARAM1 D1
3 #define PARAM1_SET(value) D1_SET(value)
```

Таблица 1.9.1: Примеры обращений к D-ячейкам

Ячейка	Чтение	Запись
D1, размер 1 байт	D1	D1_SET(<значение>)
D2, размером 2 байта	D2_W	D2_W_SET(<значение>)
Второй бит ячейки D4	D4_2	D4_2_SET(<значение>)



```
4 // Слово, состоящее из байт D2, D3
5 #define PARAM2 D2_W
6 #define PARAM2_SET(value) D2_W_SET(value)
7 // 1-й бит байта D4
8 #define PARAM3 D4_1
9 #define PARAM3_SET(value) D4_1_SET(value)
```

1.10 Ячейки статической памяти управляющих программ (R)

Обозначаются символом «GR» и могут использоваться в программе электроавтоматики для получения данных от геометрии. При каждом пуске и завершении (в том числе и аварийном) управляющей программы происходит автоматическое сохранение всех ячеек памяти на диск. Кроме того можно данную операцию можно выполнить, нажав кнопку «Сохранить файл» на вкладке «R-ячейки». На этой же вкладке доступно и редактирование каждой переменной.

1.11 Системные параметры пользователя

Системные параметры обозначаются символом "P" — для целочисленных и "PU" — для действительных параметров пользователя. Использование параметров допускается только в правой части уравнения (только для чтения). Данные параметры устанавливаются в системе параметров и им соответствуют параметры N8501–N8550 для целочисленных и N8001–N8050 для действительных значений. Назначение этих параметров произвольное и определяется разработчиком электроавтоматики. Описание назначения должно быть указано в документации на конкретный станок. Доступ к параметрам из электроавтоматики выполняется с помощью символов P (для целочисленных) либо PU (для действительных) с номером, значение которого имеет следующее соответствие: P1 — параметр N8501, PU2 — параметр N8002, P32 — параметр N8532 и т.д.

При этом каждый параметр занимает слово (2 байта).

Пример. Запись параметра в переменную

```
1 M1_W = P3; // Запись параметра N8503 в динамическую память M1_W
2 M3_W = P4 * 2; // Запись произведения параметра N8504 на число в
   динамическую память M3_W
```



```
3 M7_1 = (P5 == 2); // Запись параметра N8503 в динамическую  
   память M1_W
```

Можно также обращаться к отдельным разрядам, байтам, словам и двойным словам целочисленных параметров. При этом значения параметров при их вводе и просмотре отображаются в десятичном виде. В электроавтоматике с помощью параметров пользователя можно задавать различные станочные константы, значения которых могут меняться во время эксплуатации станка, и доступ к которым должен быть простым.

1.12 Обменные ячейки

Обменные ячейки (сигналы) служат для согласования работы CNC-модуля, интерфейса оператора и программы электроавтоматики. Взаимодействие модулей представлено на рисунке 1.12.1.

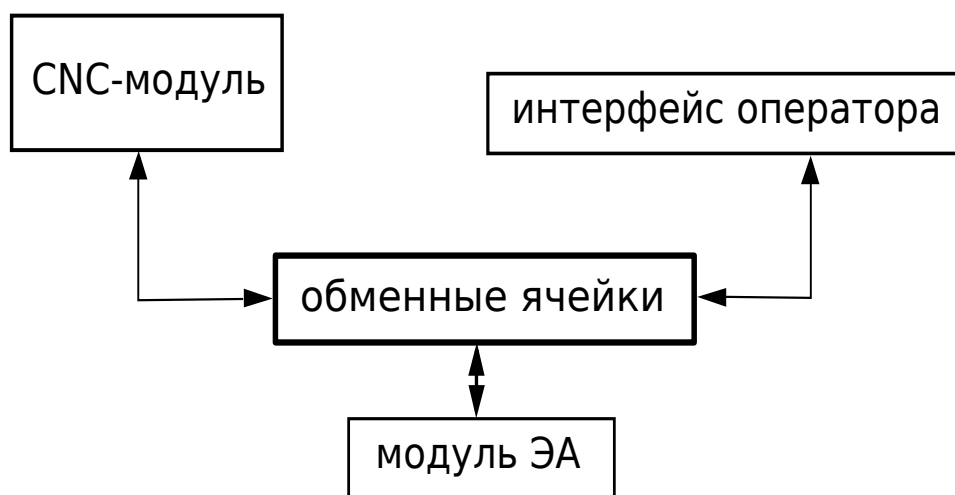


Рис. 1.12.1: Схема взаимодействия модулей

1.12.1 Общий перечень обменных ячеек

Для правильного использования обменных ячеек необходимо учитывать тип доступа к переменной, а также право на ее формирование. Сигналы типа «W» позволяют выполнять как их запись, так и чтение, и служат для передачи информации в CNC-модуль и в интерфейс оператора. Сигналы типа «R» служат для передачи информации из CNC-модуля в модуль электроавтоматики. Сигналы типа «RW» служат для двунаправленного обмена и формируются как из CNC-модуля, так и из модуля электроавтоматики. При этом нужно обращать внимание на то, какому модулю принадлежит право формирования переднего фронта этих сигналов, и какому модулю — право формирования заднего фронта этих сигналов. Права доступа указаны в описании обменных ячеек.

Обменные ячейки условно можно разделить на четыре группы:

- обменные ячейки осей;
- обменные ячейки шпинделей;
- обменные ячейки каналов;
- общие обменные ячейки.

Обменные ячейки могут быть типов: пронумерованные и именованные. Именованные ячейки зачастую являются теми же ячейками, что и пронумеро-



Таблица 1.12.1: Обозначения, определяющие доступ к обменным ячейкам

Обозначение	Пояснение
R	Ячейка доступна только на чтение.
W	Значение ячейки доступно на запись и меняется только в ПЛК.
RW	Значение ячейки доступно на запись, но может быть изменено системой ЧПУ.
I	Булева ячейка доступна только на чтение.
O	Значение булевой ячейки доступно на запись и меняется только в ПЛК.
IO	Значение булевой ячейки доступно на запись, но может быть изменено системой ЧПУ.

ванные. Однако не все именованные ячейки обязаны иметь нумерованный вариант написания.

В **именованных ячейках** используются общие сокращения, обозначающие принадлежность ячейки к той или иной задаче. Например, сокращение «АХ» в начале имени обозначает, что ячейка является осевой. А сокращение «СН» в начале имени указывает на то, что ячейка принадлежит каналу. После обозначения принадлежности ячейки в начале имени идет номер, который уточняет, к какому именно элементу ячейка относится. Например, обозначение «АХ4» обозначает, что ячейка принадлежит 4-й оси станка.

В конце каждой именованной ячейки идет одно из обозначений, описывающих возможности доступа к ячейке (см. таблицу 1.12.1).

Ячейки каналов расположены по адресам V1-V59. Интервал V1-V29 отведен под первый канал станка, а интервал V30-V59 — для второго. Значение ячеек V1-V29 полностью повторяет значение ячеек V30-V59. Список обменных ячеек для каналов приведен в таблице 1.12.2.

Пример. Соответствие обменных ячеек для разных каналов

-
- 1 CH1_M_R // код M-функции для первого канала
 - 2 CH2_M_R // код M-функции для второго канала
-

ВАЖНО: В случае одноканальной конфигурации станка обращение к обменным ячейкам других каналов ЗАПРЕЩЕНО!



Таблица 1.12.2: Обменные ячейки каналов

№	Доступ	Пояснение
	CH_COUNT_R	Количество каналов УП.
V1_W	CH1_M_R	Код M-функции.
V2_W	CH1_S_NEW_R	Код S-функции 100%.
V3_W	CH1_S_RPM_NEW_R	Заданные обороты шпинделя.
V4_W	CH1_T_N_NEW_R	Заданный номер инструмента T-функции.
V5_W	CH1_T_B_NEW_R	Заданный номер корректора T-функции.
	CH1_T_RST_TIME_O	Сбросить счётчик времени наработки выбираемого инструмента (работает только по ответу T-команды).
V6_W	CH1_T_TURR_W	Номер магазина инструментов.
V7_W	CH1_T_SLOT_W	Номер слота в магазине инструментов.
	CH1_F_FF_R	Текущая подача в УП.
	CH1_F_FB_R	Текущая подача по датчикам обратной связи осей.
	CH1_F_FF_ORIG_R	Нескорректированная подача, заданная в УП.
V8_B	CH1_F_PCT_RW	Корректор подачи F, в процентах.
V9_B	CH1_S_PCT_RW	Корректор подачи S, в процентах.
	CH1_RAPID_F_PCT_RW	Корректор подачи быстрого хода (G0), в процентах. Связанные параметры: N4005, N4006, N4007).



	CH_COUNT_R	Количество каналов УП.
V10_1	CH1_UI_SELECT_I	Запрос на установку 1-го канала текущим.
V10_2	CH1_F_PCT_IO	Разрешить коррекцию подачи.
	CH1_RAPID_F_PCT_IO	Разрешить коррекцию подачи быстрого хода (G0). Связанные параметры: N4005.
V10_3	CH1_S_PCT_IO	Разрешить коррекцию шпинделя.
V10_4	CH1_PROBE_SKIP_IO	Прервать исполнение G31 и перейти к следующему кадру.
	CH1_MOVE_PROBE_I	Идёт выполнение команды G31. Флаг выставляется параллельно с CH1_MOVE_WORKING_I.
V10_5	CH1_MOVE_RAPID_I	Выполняется перемещение на быстром ходу.
V10_6	CH1_MOVE_WORKING_I	Выполняется перемещение на рабочей подаче.
V12_1	CH1_M_I	Будет выдача М-функции.
V12_2	CH1_M_O	Ответ по М-функции.
V12_3	CH1_S_I	Будет выдача S-функции.
V12_4	CH1_S_O	Ответ S-функции.
V12_5	CH1_T_I	Будет выдача T-функции.
V12_6	CH1_T_O	Ответ T-функции.
V12_7	CH1_MAN_T_O	Применение инструмента в ручном режиме.
V12_8	CH1_MP_IO	Будет выдача параллельной М-функции.
V13_1	CH1_SCAN_ADD_FRAME_OI	Запись кадра в режиме обучения.



	CH_COUNT_R	Количество каналов УП.
V13_2	CH1_SCAN_ADD_FRAME_I	Флаг готовности графического интерфейса к записи кадра в режиме обучения.
V13_3	CH1_SCAN_SET_ON_IO	Включение режима обучения.
V13_4	CH1_SCAN_ON_I	Режим обучения включен.
V13_5	CH1_SCAN_UI_ON_I	Запрос на включение режима обучения от графического интерфейса.
V13_6	CH1_SCAN_UI_OFF_I	Запрос на выключение режима обучения от графического интерфейса.
V13_7	CH1_SCAN_UI_ADD_FRAME_I	Запрос на запись кадра в режиме обучения от графического интерфейса.
	CH1_SCAN_G_OI	G-кадр готов к записи в режиме обучения.
	CH1_SCAN_G_CODE_W	Номер G-кода.
	CH1_SCAN_G_MOD_W	Модификатор G-кода.
	CH1_SCAN_M_OI	M-функция готова к записи в режиме обучения.
	CH1_SCAN_M_CODE_W	Номер M-функции.
	CH1_SCAN_M_MOD_W	Модификатор M-функции.
	CH1_SCAN_F_AUTO_OI	Автоматическое вычисление подачи в режиме обучения.
	CH1_SCAN_F_OI	Задана подача в режиме обучения.
	CH1_SCAN_F_VAL_W	Значение подачи в режиме обучения.
	CH1_SCAN_T_OI	T-команда готова к записи в режиме обучения.



	CH_COUNT_R	Количество каналов УП.
	CH1_SCAN_T_N_W	Номер инструмента в режиме обучения.
	CH1_SCAN_T_B_W	Номер привязки инструмента в режиме обучения.
V13_8	CH1_JOG_O	Запрос на включение режима толчка.
V14_1	CH1_MAN_O	Запрос на включение ручного режим.
V14_2	CH1_WHL_O	Запрос на включение режима управления осями от маховика.
V14_3	CH1_PULSE_O	Запрос на включение режима импульсных перемещений.
V14_4	CH1_HOME_O	Запрос на включение режима выхода в 0.
V14_5	CH1_STEP_O	Запрос на включение покадрового режима.
V14_6	CH1_AUTO_O	Запрос на включение автоматического режима.
V14_7	CH1_MDI_O	Запрос на включение режима предварительного набора.
V14_8	CH1_MDI_I	Режим MDI включен.
V15_1	CH1_UI_MAN_IO	Запрос на переход в ручной режим от графического интерфейса к ПЛК.
V15_2	CH1_STEP_I	Включен пошаговый режим.
V15_3	CH1_AUTO_I	Включен автоматический режим.
V15_4	CH1_UI_HOME_IO	Запрос на переход в режим выхода в 0 от графического интерфейса к ПЛК.



	CH_COUNT_R	Количество каналов УП.
V15_5	CH1_UI_STEP_IO	Запрос на переход в покадровый режим от графического интерфейса к ПЛК.
V15_6	CH1_UI_AUTO_IO	Запрос на переход в автоматический режим от графического интерфейса к ПЛК.
V15_7	CH1_UI_MDI_IO	Запрос на переход в режим MDI от графического интерфейса к ПЛК.
V16_1	CH1_PRG_RUN_IO	Пуск УП. Выставляется на один такт, сбрасывается автоматически при запуске УП.
V16_2	CH1_PRG_STOP_IO	Приостановка УП. Пока данная ячейка выставлена, УП будет в состоянии приостановки.
V16_3	CH1_PRG_RUNNIG_I	Флаг исполнения УП. Выставляется в момент запуска, сбрасывается по окончании или отмене исполнения.
V16_4	CH1_PRG_ABRT_O	Отмена исполнения УП. Достаточно выставить минимум на 1 такт. Сбрасывается автоматически по сбросу программы.
	CH1_PRG_ABRT_IO	УП была отменена.
V16_5	CH1_PRG_FIN_O	Завершить УП по M-функции.
	CH1_PRG_FIN_IO	УП была завершена.
V16_6	CH1_PRG_STOP_I	УП приостановлена.



	CH_COUNT_R	Количество каналов УП.
V16_7	CH1_PRG_CHOSEN_I	УП выбрана и может быть исполнена в автоматическом режиме.
V16_8	CH1_PRG_RELOAD_O	Повторный выбор УП по М-команде.
V17_1	CH1_UI_PRG_RUN_IO	Запрос пуска УП от графического интерфейса к ПЛК.
V17_2	CH1_UI_PRG_STOP_IO	Запрос приостановки УП от графического интерфейса к ПЛК.
V17_3	CH1_UI_PRG_ABRT_IO	Запрос отмены исполнения УП от графического интерфейса к ПЛК.
V17_7	CH1_PRG_STARTED_I	Получен запрос на старт УП от ПЛК, но интерполятор ещё не приступил к исполнению.
V17_8	CH1_CAN_STOP_I	Разрешение останова УП.
V18_1	CH1_JOG_I	Включен режим безразмерных перемещений.
V18_2	CH1_MAN_I	Включен ручной режим управления.
V18_3	CH1_WHL_I	Включен режим управления от маховика.
V18_4	CH1_PULSE_I	Включен режим импульсных перемещений.
V18_5	CH1_HOME_I	Включен режим выход в 0.
V18_7	CH1_COND_STOP_IO	Включен условный останов по M1.
V19_1	CH1_SP_BY_PLC_O	Разрешить управление текущим шпинделем от ПЛК.



	CH_COUNT_R	Количество каналов УП.
V19_3	CH1_SP_IS_AX_IO	Текущий шпиндель в режиме оси.
V20_W	CH1_SP_VAL_W	Значение на ЦАП шпинделя.
V22_W	CH1_T_ID_W	Идентификатор текущего инструмента.
	CH1_T_OFFSETS_ID_W	Идентификатор текущей привязки инструмента.
V23_W	CH1_F_LIM_RW	Ограничение скорости.
V24_W	CH1_T_BIND_W	Значение корректора T для индикации.
V25_W	CH1_UI_S_RPM_W	Скорость вращения текущего шпинделя.
V26_W	CH1_SP_GEAR_W	Номер коробки передач текущего шпинделя.
V27_W	CH1_SP_PULSES_R	Количество импульсов шпинделя на оборот.
V28_W	CH1_MP_R	Параллельный M-код.
V29_W	CH1_SP_N_NEW_R	Номер текущего шпинделя, для которого поступило задание из УП.
	CH1_PART_COUNTER_W	Счетчик деталей.

Осевые обменные ячейки расположены по адресам V100–V900. Интервал V100–V199 отведен под первую ось, интервал V200–V299 — под вторую, и т.д. Список обменных ячеек шпинделей приведен в таблице 1.12.4.

Пример. Соответствие обменных ячеек для разных осей

-
- 1 AX1_MCS_POS_R // абсолютное положение оси 1
 - 2 AX2_MCS_POS_R // абсолютное положение оси 2
 - 3 AX3_MCS_POS_R // абсолютное положение оси 3
-

ВАЖНО: ЗАПРЕЩАЕТСЯ обращаться к ячейкам осей, номер которых превышает количество осей, прописанных в конфигурации станка (па-



раметр N1000)! Например, при $N1000 = 3$ запрещается обращаться к AX4_REMAINS_R.

Таблица 1.12.4: Осевые обменные ячейки

№	Имя ячейки	Описание
	AX_COUNT_R	Количество осей.
V100_W	AX1_MCS_POS_R	Абсолютное положение оси.
V101_W	AX1_PCS_POS_R	Рабочее положение оси.
	AX1_BW_POS_R	Позиция оси по обратной связи.
V102_W	AX1_REMAINS_R	Остаток перемещения в кадре.
V105_W	AX1_MAN_POS_W	Целевая координата при движении в ручном режиме.
V106_W	AX1_RPM_AVG_R	Скорость вращения в оборотах в минуту.
	AX1_F_GIVEN_FF_R	Текущая заданная скорость оси.
V107_W	AX1_F_FF_R	Текущая скорость оси с учётом PID-регулятора.
V108_W	AX1_F_FB_R	Текущая скорость оси (реальная).
V109_W	AX1_POS_ERROR_R	Рассогласование оси.
V110_W	AX1_MAN_F_W	Заданная подача в ручном режиме.
V111_W	AX1_MAN_WHL_STEP_W	Дискретность маховиков.
V112_W	AX1_HOME_POS_NEW_W	Новая позиция для координаты, записанной в AX1_HOME_POS_RW (для выхода в 0).



V113_W	AX1_HOME_POS_RW	Позиция «0» станка, относительно которой назначается новая позиция нуля из ячейки AX1_HOME_POS_NEW_W (для выхода в 0).
V114_W	AX1_LOAD_VAL_W	Нагрузка на ось (0-255), пишем в ПЛК.
V115_W	AX1_MAN_WHL_N_W	Номер маховика для оси.
V120_1	AX1_MOVE_ON_O	Разрешение перемещения оси.
V120_2	AX1_PID_ON_O	Включение слежения за координатой.
V120_3	AX1_GRAB_O	Запрос на включение режима перехвата управления осью от ПЛК.
V120_4	AX1_GRAB_I	Режим управления осью от ПЛК включен.
V120_5	AX1_MOVE_SUPPRESS_R	Движение было перехвачено ПИД-регулятором.
V120_6	AX1_GEAR_O	Установить новый диапазон скоростей.
V120_7	AX1_GRAB_POS_O	Команда на перемещение в позицию, заданную ячейкой AX1_GRAB_POS_W.
V120_8	AX1_GRAB_POS_IO	Достигнута координата в режиме от ПЛК.
V121_1	AX1_LIM_MINUS_O	Ограничение движения в «-».
V121_2	AX1_LIM_PLUS_O	Ограничение движения в «+».
V121_3	AX1_LIM_HOME_O	Сработал концевик выхода в «0».
V121_4	AX1_LIM_F_MINUS_O	Снижение скорости «+».
V121_5	AX1_LIM_F_PLUS_O	Снижение скорости «-».



V121_6	AX1_HOMING_IO	Выход в «0» завершен.
V121_7	AX1_PID_ERR_IO	Ошибка ПИД-регулятора. Для продолжения управления осью необходимо сбросить в 0.
V121_8	AX1_PID_RESET_O	Сброс ошибки ПИД-регулятора (сброс рассогласования).
	AX1_DRIFT_ON_O	Включение дрейфа приводов по оси.
V122_1	AX1_MOVE_ON_I	Будет движение по оси.
V122_2	AX1_LIM_HOME_I	Был выход в «0».
V122_3	AX1_MOVE_MINUS_I	Идет перемещение в «-».
V122_4	AX1_MOVE_PLUS_I	Идет перемещение в «+».
V122_5	AX1_MOVE_ZONE_I	Ось в зоне контроля.
V122_7	AX1_TRAP_ZERO_O	Начать поиск 0-метки.
V122_8	AX1_TRAP_ZERO_IO	Найдена 0-метка.
V123_1	AX1_MAN_GO_PLUS_O	Движение в «+» в ручном режиме.
V123_2	AX1_MAN_GO_MINUS_O	Движение в «-» в ручном режиме.
V123_3	AX1_MAN_F_O	Задана подача в ручном режиме.
V123_4	AX1_NEW_POS_O	Установить новую позицию для координаты, записанной в ячейку AX1_HOME_POS_RW.
V123_5	AX1_NEW_POS_IO	Новая позиция из ячейки AX1_NEW_POS_O установлена.
V123_6	AX1_HOME_I	Начат выход в «0».
V123_7	AX1_MOVE_WARN_R	Скорость была принудительно изменена или ограничена.
V123_8	AX1_MAN_MOVE_RST_O	Сброс последней операции перемещения.



V124_1	AX1_DRV_OUT_O	Включить управление приводом (разрешить задание на ЦАП).
V124_2	AX1_DRV_ON_O	Включить привод (для цифровых приводов).
V124_3	AX1_DRV_OP_O	Разрешить выполнение операции на приводе для текущего режима (для цифровых приводов).
V124_4	AX1_DRV_ON_I	Привод включен (для цифровых приводов).
V124_5	AX1_DRV_OP_I	Разрешено выполнять операции на приводе (для цифровых приводов).
V124_8	AX1_MAN_POS_O	Целевая координата для перемещения была выставлена.
V125_W	AX1_GRAB_F_W	Скорость позиционирования оси в режиме перехвата управления от ПЛК.
V126_W	AX1_GRAB_POS_W	Целевая координата (в машинной системе координат), в которую необходимо переместиться в режиме перехвата управления от ПЛК.
V127_W	AX1_GEAR_W	Номер выставяемого диапазона скоростей.
V130_W	AX1_MAN_WHL_F_W	Скорость перемещения по маховику.



V131_W	AX1_DRV_MODE_R	Текущий режим работы привода (для цифровых приводов): «0» — неизвестный режим; «3» — управление скоростью; «4» — управление положением; «5» — управление моментом; «6» — выход в «0»; «7» — интерполируемое управление положением; «8» — циклическое управление позицией; «9» — циклическое управление скоростью; «10» — циклическое управление моментом.
V132_W	AX1_DRV_MODE_W	Установить режим работы привода (для цифровых приводов): «0» — неизвестный режим; «3» — управление скоростью; «4» — управление положением; «5» — управление моментом; «6» — выход в «0»; «7» — интерполируемое управление положением; «8» — циклическое управление позицией; «9» — циклическое управление скоростью; «10» — циклическое управление моментом.



V133_W	AX1_DRV_STATE_R	Текущее состояние привода (для цифровых приводов): «0» — не готов к включению; «1» — включение невозможно; «2» — готов к включению; «3» — включен; «4» — разрешено исполнение задания; «5» — быстрый останов; «6» — останов по ошибке; «7» — состояние ошибки.
V135_W	AX1_DRV_TORQUE_R	Текущий момент привода (для цифровых приводов).
V136_W	AX1_DRV_POS_ERR_R	Текущее рассогласование привода (для цифровых приводов).
V137_W	AX1_DRV_RPM_R	Текущая скорость привода (для цифровых приводов).
V140_1	AX1_DRV_REACHED_R	Последнее задание на привод было исполнено (для цифровых приводов).
V140_2	AX1_DRV_ZERO_SPD_R	Скорость привода равна нулю (для цифровых приводов в режиме управления по скорости).
V140_8	AX1_DRV_HALT_W	Команда на аварийный останов двигателя (для цифровых приводов).
V141_1	AX1_INACTIVE_O	Сделать ось неактивной (скрыть из графического интерфейса).
V141_2	AX1_INACTIVE_I	Ось является неактивной.



V141_3	AX1_MODE_SLAVE_O	Включить режим подчинённой оси, в котором ось будет перемещаться вместе с главной осью, заданной в параметрах.
V141_5	AX1_GRAB_ALLOW_F_K_O	Запрос на разрешение процентовки скорости перемещения оси в режиме перехвата управления осью.
V141_6	AX1_GRAB_ALLOW_F_K_I	В режиме перехвата управления осью разрешена процентовка скорости перемещения. Из ПЛК запрашивается через ячейку AX1_GRAB_ALLOW_F_K_O. Значение может временно изменяться системой ЧПУ при возврате на контур УП после перемещения в ручном режиме.
V141_7	AX1_GRAB_GO_PLUS_IO	Команда на движение в плюс в режиме перехвата управления от ПЛК.
V141_8	AX1_GRAB_GO_MINUS_IO	Команда на движение в минус в режиме перехвата управления от ПЛК.
V142_1	AX1_HOME_O	Начать выход в «0» по оси.
V142_2	AX1_HOME_QRY_IO	Запрос на выход в «0» по оси.
V142_3	AX1_HOME_OK_IO	Выход в «0» по оси был завершён.
V142_4	AX1_HOME_RST_IO	Отменить выход в «0» по оси.
V142_5	AX1_HOME_ABORTED_IO	Выход в «0» по оси был отменён.



V142_6	AX1_POS_LIM_MUNIS_I	Ось наехала на программный ограничитель в «минус».
V142_7	AX1_POS_LIM_PLUS_I	Ось наехала на программный ограничитель в «плюс».

Обменные ячейки шпинделей доступны только в виде именованных ячеек. Список ячеек, относящихся к шпинделям, приведен в таблице 1.12.6.

Обменные ячейки суппортов доступны только в виде именованных ячеек. Список ячеек, относящихся к суппортам, приведен в таблице 1.12.7.

Общие обменные ячейки расположены по адресам V60-V99.



Таблица 1.12.6: Обменные ячейки шпинделей

Имя ячейки	Описание
SP_COUNT_R	Количество шпинделей.
SP1_OUT_W	Управляющее задание на привод.
SP1_GEAR_W	Номер передачи коробки скоростей.
SP1_BY_PLC_O	Управление шпинделем от ПЛК.
SP1_IS_AX_IO	Шпиндель в режиме оси.
SP1_AX_N_R	Номер оси шпинделя.
SP1_RPM_R	Текущие обороты шпинделя.
SP1_UI_RPM_O	Отображать на экране обороты из SP1_UI_RPM_W при отсутствии ДОС.
SP1_UI_RPM_W	Отображаемые на экране обороты при отсутствии ДОС.

Таблица 1.12.7: Обменные ячейки суппортов

Имя ячейки	Описание
SUP_COUNT_R	Количество суппортов.
SUP1_TOOL_D_W	Коррекция диаметра инструмента, используемого в данном суппорте.
SUP1_T_WEAR_SEC_R	Время наработки текущего инструмента в секундах.
SUP1_T_WEAR_TIME_OVER_I	Превышено ли время наработки на отказ у текущего инструмента.
SUP1_T_WEAR_DISABLE_O	Запретить подсчёт времени наработки инструмента (для холостых перемещений).
SUP1_T_N_R	Текущий номер инструмента.
SUP1_T_OFFSETS_R	Текущий номер привязки.
SUP1_T_ID_R	Идентификатор текущего инструмента.
SUP1_T_OFFSETS_ID_R	Идентификатор текущей привязки инструмента.



Таблица 1.12.8: Общие обменные ячейки

№	Имя ячейки	Описание
V60_1	MACH_UI_RDY_I	Готовность ЧПУ.
V60_2	MACH_NO_ERR_O	Отсутствие аварии: «0» — произошла авария.
V60_5	MACH_DRV_ON_O	Привода включены.
V60_6	MACH_ON_O	Станок включен.
V60_7	MACH_DRIFT_ON_O	Разрешить коррекцию дрейфа.
V61_1	UI_KEY_DOWN_I	Нажато сочетание клавиш из UI.
V61_4	MACH_UI_ESC_IO	Клавиша ESC.
V61_5	MACH_UI_CL_DLG_IO	Закрытие диалогового окна
V61_7	MACH_DRV_ERR_IO	Ошибка рассогласования по осям.
V62_5	MACH_UI_ERR_RST_IO	Сброс вибраторов ошибок (от ЧПУ).
V62_6	MACH_ERR_RST_O	Сброс вибраторов ошибок (от ПЛК).
V63_1	UI_MOD_CTRL_I	В UI зажата клавиша Ctrl.
V63_2	UI_MOD_ALT_I	В UI зажата клавиша Alt.
V63_3	UI_MOD_SHIFT_I	В UI зажата клавиша Shift.
V63_4	UI_MOD_SUPER_I	В UI зажата клавиша Super.
V63_6	UI_MOD_HYPER_I	В UI зажата клавиша Hyper.
V69_B	MACH_WHL_STEP_W	Шаг энкодеров (множитель).
V70_B	MACH_CH_N_RW	Текущий канал.
V71_B	UI_KEY_ALIAS_R	Условный код сочетания клавиш из UI.
	UI_PERM_MODE_R	Текущий режим прав доступа (EAUTO_PERM_OPERATOR — оператор, EAUTO_PERM_PROGRAMMER — программист, EAUTO_PERM_ENGINEER — наладчик, EAUTO_PERM_ADMIN — администратор).
	UI_PERM_MODE_W	Команда на смену режима прав доступа (EAUTO_PERM_NONE — нет запроса, остальное аналогично UI_PERM_MODE_R, но смена на EAUTO_PERM_ADMIN запрещена).



1.13 Таймеры

В программе могут использоваться таймеры. Таймер может использоваться как в левой, так и в правой частях уравнения. Символ «Т» с номером используется для задания установки времени последующего срабатывания этого таймера в программе электроавтоматики. Поэтому символ «Т» с номером может использоваться только в левой части уравнения. Время задается в количестве итераций секции, в которой таймер будет использоваться.

Пример. Задание значения таймеру

```
1 T14 = 10; // Таймер 14 сработает через 10 повторений секции
  (быстрой или медленной) при условии, что на входе таймера 1
  (для TS);
2 T15 = P1*10; // Сигнал на выходе появится через количество
  итераций, которое определит результат в правой части
  уравнения (для TS);
```

Время срабатывания таймера можно рассчитать по формуле: $T = n * t$, где n — количество итераций секции, заданное для таймера; t — время итерации секции.

Пример. Расчет времени срабатывания таймера

```
1 T1 = 5;
```

Например, $N110 = 100$ мс, значит, на выходе таймера появится единица через 500 мс ($5*100$) при условиях, что на вход таймера подано значение 1, и таймер типа TS (при использовании таймера типа TR сигнал на выходе сбросится через заданное время).

Существуют два варианта использования таймеров в программе электроавтоматики: прямые и инверсные.

Для **прямых** таймеров (TS):

- на вход подаётся 1 для включения таймера;
- на выходе выдаётся 0, пока таймер не досчитает до требуемого значения;
- на выходе выставляется 1, как только таймер досчитает до требуемого значения.



Для **инверсных** таймеров (TR):

- на вход подаётся 0 для включения таймера;
- на выходе выдаётся 1, пока таймер не досчитает до требуемого значения;
- на выходе выставляется 0, как только таймер досчитает до требуемого значения.

Сброс таймер можно сделать с помощью команды **T<номер>R(1)**. Команда **T<номер>R(0)** не осуществляет никаких действий.

Пример. Сброс таймера по его срабатыванию

```
1 U1_1 = U1_1 ^ TOS1; // переключение по срабатыванию таймера
2 TR(TOS1); // сбрасываем, если таймер сработал
```

Допустимое максимальное значение таймера — 32,767.

1.13.1 Таймер задержки установки

Первый вид таймера задается символами **TS** с номером и может использоваться в левой части уравнения. Этот таймер, стоящий в левой части уравнения (TS), указывает на формирование входного сигнала этого таймера, а таймер, стоящий в правой части уравнения (обозначается TOS), указывает на использование выходного сигнала таймера.

Принцип работы такого таймера заключается в следующем: до тех пор, пока на входе есть ненулевой сигнал, таймер производит подсчет времени. При достижении заданного значения (указанного в заголовке программы через символ T с номером) включается выходной сигнал таймера. Если в процессе счета на входе таймера сигнал станет равным нулю, то выходной сигнал таймера сбрасывается, и счет начинается сначала. Необходимо учитывать, что таймеры считают не время, а итерации секций. При изменении параметра N109 (N110) необходимо соответствующим образом изменять все установки таймеров для сохранения правильности расчета времени.

Пример. Таймер задержки включения

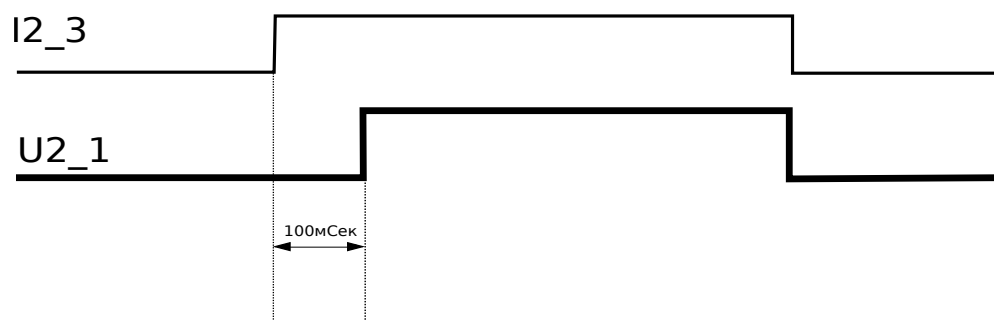


Рис. 1.13.1: Диаграмма срабатывания таймера типа TS

```
1 PLC_VAR_INIT_STARTT
2 1 = 100;
3 PLC_VAR_INIT_END
4 PLC_SLOW_START
5 TS1(I2_3);
6 U2_1 = TOS1;
7 PLC_SLOW_END
```

В данном примере выход U2_1 включится через 100, если вход I2_3 все это время будет включен, и будет выключен, если вход I2_3 будет выключен. Диаграмма срабатывания сигналов приведена на рисунке 1.13.1.

1.13.2 Таймер задержки сброса

Второй вид таймера задается символами **TR (TOR)** с номером и может использоваться как в левой, так и в правой частях уравнения. При этом таймер, стоящий в левой части уравнения (TR), указывает на формирование входного сигнала этого таймера, а таймер, стоящий в правой части уравнения (TOR), указывает на использование выходного сигнала таймера.

Принцип работы такого таймера заключается в следующем: выходной сигнал таймера устанавливается сразу же, как только появляется входной сигнал. При сбросе входного сигнала таймер производит подсчет времени и сохраняет установленное состояние выходного сигнала. При достижении заданного значения (указанного в заголовке программы через символ T с номером) выключается выходной сигнал таймера.

Пример. Таймер задержки сброса

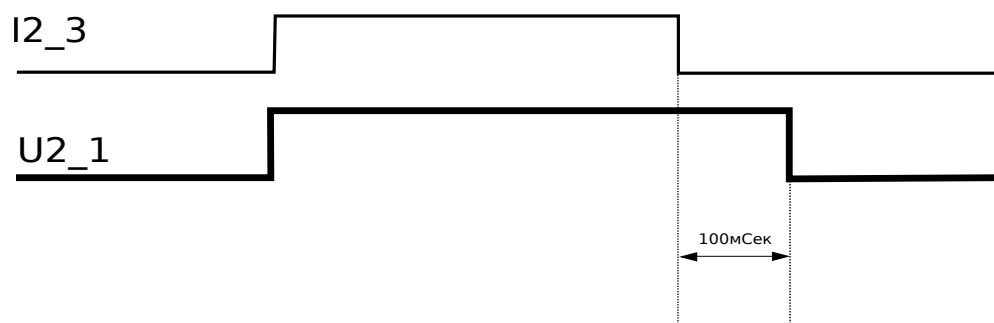


Рис. 1.13.2: Диаграмма срабатывания сигналов TR таймера

```
1 PLC_VAR_INIT_STARTT
2 1 = 100;
3 PLC_VAR_INIT_END
4
5 PLC_SLOW_START
6 TR1(I2_3);
7 U2_1 = TOR1;
8 PLC_SLOW_END
```

В данном примере выход U2_1 включится сразу же при включении входа I2_3 и выключится через 100 итераций медленной секции после выключения выхода I2_3.

Диаграмма срабатывания сигналов приведена на рисунке 1.13.2.

Задание установки таймера возможно не только в инициализации программы электроавтоматики, но и в других ее частях. При этом нужно иметь в виду, что, если в каждом такте программы будет происходить ее прохождение через задание установки таймера, такой таймер никогда не сработает, поэтому необходима организация установки только при их изменении.

Пример. Задание величины таймера через условие

```
1 if (P1_W == M1_W) {
2 T1 = P1_W;
3 M1_W = P1;
4 }
```

В примере установка таймера T1 изменяется только при изменении целочисленного параметра пользователя N8501 (P1W).



1.14 Счетчики

Счетчик обозначается символом «С». Счетчик может использоваться как в левой, так и в правой частях уравнения.

Счетчик имеет два входа (прибавляющий и вычитающий), обозначаемых C1_I и C1_D соответственно. Входы счетчика используются только в левой части уравнения. Счетчик производит подсчет импульсов (по переднему фронту входного сигнала). При превышении заданного значения, указанного в заголовке программы (в случае использования прибавляющего входа счетчика), значение счетчика (C_OUT) устанавливается в 1. При вычитании при достижении 0 (в случае использования вычитающего входа счетчика) счетчик устанавливается в заданное значение (C_VAL), указанное в заголовке программы. Допускается изменять значение счетчика в процессе выполнения программы прямым присвоением. Значение счетчика может использоваться как в логических уравнениях, так и в арифметических выражениях. Сбросить текущую величину счетчика можно, подав 1 на вход C_RESET.

Пример. Варианты использования счетчиков

```
1 PLC_VAR_INIT_STARTC
2 V1 = 100;C
3 V2 = 200;C
4 V3 = 1500;
5 PLC_VAR_INIT_END
6
7 PLC_SLOW_START
8 CI1(I2_3);
9 CD2(I1_4);C
10 R2 = (C2 == 32) && I1_1;
11 U2_2 = C(1 > 50);
12 PLC_SLOW_END
```

В данном примере счетчик C1 считает количество включений сигнала I2_3, счетчик C2 работает на вычитание (количество включений I1_4), а значение счетчика C3 будет всегда равно числу 345. Если значение счетчика C1 больше 50, то выход U2_2 будет включен. Счетчик C2 будет сброшен (C = 0) в случае, если на вход I1_1 подана 1, а его значение при этом равно 32.

Допустимое максимальное значение счётчика — 32,767.

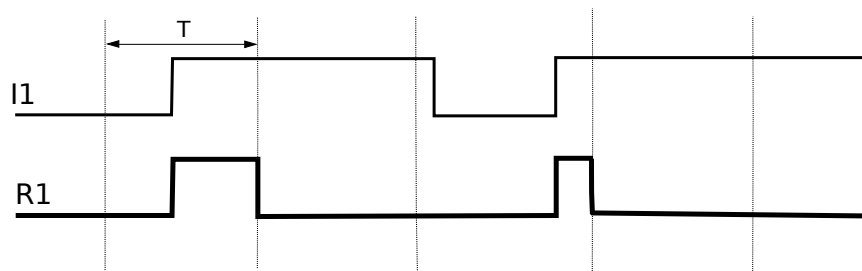


Рис. 1.15.1: Диаграмма работы одновибратора

1.15 Одновибраторы

Одновибраторы или генераторы одиночных импульсов представляют собой вид промежуточной памяти, которая устанавливается в единичное состояние в том случае, если состояние правой части уравнения, формирующего одновибратор, перешло из нулевого состояния в единичное (или наоборот). Таким образом, с помощью одновибраторов реализуется задача фиксации переднего (заднего) фронта каких-либо сигналов или значений выражений в логических уравнениях. При этом единичное состояние одновибратора сохраняется только на один проход быстрой или медленной секции программы электроавтоматики.

В левой части уравнения используется вход одновибратора (RI) с соответствующим номером. В правых частях уравнения и функциях используется ТОЛЬКО выход вибратора (R)!

Пример. Пример работы одновибратора

```
1 RI1 = I2_1;
```

Диаграмма срабатывания сигналов приведена на рисунке 1.15.1.

ВАЖНО: Момент сброса состояния одновибраторов зависит от того, в какой секции программы (и ее части) формируются одновибраторы. Запрещается размещать вход и выход одновибратора в разных секциях программы.

Для того, чтобы одновибратор не отреагировал на первый передний фронт сигнала, нужно в инициализации программы электроавтоматики установить единичное значение одновибратора.

Пример. Пропуск первого срабатывания одновибратора



```
1 RI1 = !I1_5;  
2 U1_2 = (R1 || U1_2) && !I2_1;
```

В данном случае выход U1_2 должен включиться по заднему фронту входа I1_5 и выключиться по переднему фронту входа I2_1. Но если, например, при первом проходе программы электроавтоматики входные сигналы еще не имеют питания, выход U1_2 включится. Чтобы избежать этого, достаточно сделать следующее:

```
1 PLC_VAR_INIT_START  
2 RI1 = 1;  
3 PLC_VAR_INIT_END
```

1.16 Специальные функции и переменные

В программе электроавтоматики могут использоваться следующие специальные функции:

- функция «выхода в 0» по оси;
- функция прекращения «выхода в 0» (по всем осям);
- формирование сообщений и ошибок;
- функции и переменные для работы с программируемой виртуальной клавиатурой;
- функция поиска в таблице инструментов;
- функция управления осью от ПЛК;
- функция привязки текущего инструмента от ПЛК;
- функция задания смещения от ПЛК;
- функция задания коррекции смещения от ПЛК;
- функция увеличения коррекции смещения от ПЛК;
- функция задания высоты текущего инструмента от ПЛК;
- функция задания высоты корректора от ПЛК;



- функция получения номера слота;
- функция получения номера инструмента в слоте;
- функция вставки инструмента в слот;
- функция получения количества магазинов инструментов;
- функция получения кода ошибки магазина инструментов;
- функция получения типа магазина инструментов;
- функция получения количества слотов в магазине инструментов;
- функция получения номера слота в магазине инструментов по его идентификатору;
- функция получения идентификатора слота в магазине инструментов по его номеру;
- функция получения идентификатора инструмента по номеру слота в магазине инструментов;
- функция получения координат инструмента по номеру слота в магазине инструментов;
- функции получения/записи битов переменной;
- функции режима обучения;
- функции создания и работы с пультом быстрых программ.

Использование всех специальных функции НЕ ОБЯЗАТЕЛЬНО, однако это будет полезно для упрощения разработки ПЛК-модуля.

1.16.1 Функция «выхода в 0» по оси

Синтаксис:

HOMING_START(<УсловиеПлюс>, <УсловиеМинус>, <НомерОси>);

В аргументах функции указываются условия выхода в ноль (минус/плюс) и номер оси, по которой этот выход будет осуществлен.

ВАЖНО: Для работы функции необходимо, чтобы был выбран соответствующий режим (CH1_HOME_0 в конфигурации с одним каналом)!

Пример. Выход в ноль по оси 1



```
1 HOMING_START(SB_X_MINUS && CH1_HOME_I, SB_X_PLUS && CH1_HOME_I,  
  1);  
2 // SB_X_MINUS, SB_X_PLUS – кнопки X+ и X- на пульте оператора
```

ВАЖНО: Выход в ноль возможно остановить только с помощью функции `HOMING_RESET` (см 1.16.2), причем по всем осям сразу!

1.16.2 Функция прекращения «выхода в 0»

Синтаксис:

```
HOMING_RESET(<Условие>);
```

В скобках следует указать условие, по которому будет прекращен выход в ноль.

ВАЖНО: Прекращение выхода в ноль будет осуществлено по всем осям.

Пример. Остановка выхода в ноль

```
1 HOMING_RESET(SB_RESET && CH1_HOME_I); // Остановить выход в  
  ноль, если в соответствующем режиме нажали кнопку сброса  
  (SB_RESET) или выключили режим выхода в ноль
```

1.16.3 Формирование сообщений и ошибок

Синтаксис:

```
ERR(<Условие>, N, "ТЕКСТ ОШИБКИ");  
VIEW(<Условие>, N, "ТЕКСТ СООБЩЕНИЯ");  
DIALOG(<Условие>, N, "ТЕКСТ СООБЩЕНИЯ", "ЦВЕТ");
```

Функции **ERR()** и **VIEW()** работают одинаково: при соблюдении условия (<Условие>) формируется ошибка/сообщение (**ERR/VIEW**) под номером **N** с текстом, указанным в функции. Разница между сообщением (**VIEW**) и ошибкой (**ERR**) только в способе отображения их в интерфейсе пользователя. **DIALOG()**, помимо записи сообщения в журнал, вызывает диалоговое окно с текстом ошибки и позволяет указать цвет ошибки в текстовом виде («red» — красный, «green» — зелёный, «blue» — синий, «orange» — оранжевый).

ВАЖНО: Система ошибок/сообщений работает на основе одновибраторов, т. е. сообщения формируются **ОДНОКРАТНО** при смене условия



(0 -> 1)! Сообщение/ошибка останется на экране до формирования более нового сообщения/ошибки.

ВАЖНО: Очистка поля ошибок происходит через обменную ячейку MACH_ERR_RST_O.

Пример. Формирование и очистка ошибки

```
1 ERR(ErrorDriver, 1, "Ошибка включения приводов подач.");
2 ERR(ErrStanokOn, 2, "Ошибка включения станка.");
3 Dialog(ErrEmergencyStop, 3, "Нажат грибок аварийного выключения
  станка.")
4 RI250 = SB_RESET || Key_ESC;
5 MACH_ERR_RST_O = R250;
```

В примере ошибки под номерами 1 и 2 будут сформированы при появлении 1 в сигналах ErrorDriver или ErrStanokOn. Диалог нажатия аварийного грибка будет показан при формировании сигнала ErrEmergencyStop. Очистка поля ошибок будет осуществлена по нажатию на кнопки сброса SB_RESET на пульте оператора или клавишу Esc на клавиатуре.

1.16.4 Работа с программируемой виртуальной клавиатурой

Синтаксис:

```
// Инициализация кнопки в позиции (1,1)
KEY(1, 1, NULL, "auto");
// Кнопка в позиции (1,1)
SB1_1
// «Диод» кнопки в позиции (1,1)
HL1_1
```

В функции первые два аргумента — строка и столбец для кнопки, затем желаемый текст для нее и иконка. Иконка для кнопки может находиться в любой из папок для иконок в каталоге «/home/titanium/icons/». Если не требуется указание иконки/текста для кнопки, в аргументе указывается служебное слово **NULL**.

Размерность поля клавиатуры указывается с помощью параметра (532/533).

Работа с кнопкой ничем не отличается от работы с клавишами пульта. Обращение к кнопке — SB, к диоду — HL с соответствующими номерами.



1.16.5 Функция поиска в таблице инструментов

Синтаксис:

```
<Ответ> = SELECT_TOOL_AND_OFFSETS (<ПодтверждениеВыбора>,
                                     <НомКанала>,
                                     <НомИнструмента>,
                                     <НомПривязки>);
```

Данная вспомогательная функция проверяет наличие записи в таблице инструментов по запрошенному номеру инструмента (**<НомИнструмента>**) и привязки (**<НомПривязки>**) для соответствующего канала (**<НомКанала>**). Для удобства функция вернёт **<ПодтверждениеВыбора>**, что можно использовать для ответа по T-команде. Функция запишет в **CH1_T_ID_W** и **CH1_T_OFFSETS_ID_W** значения, соответствующие записям в таблице инструмента. Если инструмент не найден, в **CH1_T_ID_W** будет -1, если привязка не найдена, в **CH1_T_OFFSETS_ID_W** будет записано -1.

Пример. Функция поиска инструмента

```
1 RI19 = CH1_T_I;
2 CH1_T_0 = SELECT_TOOL_AND_OFFSETS(R19, 1, CH1_T_N_NEW_R,
    CH1_T_B_NEW_R);
```

1.16.6 Функция управления осью от ПЛК

Синтаксис:

```
AXIS_CONTROL(<on>, <off>, <num>, <pos>, <speed>);
```

ВАЖНО: при использовании данной функции ось автоматически переводится в режим перехвата, а работа ячейки включения управления от ПЛК блокируется ($Vn20_3$, где **n** — номер оси).

В аргументах функции указываем номер оси (**<num>**), позицию (**<pos>**) точки в машинных координатах для перемещения и скорость (**<speed>**), на которой перемещение будет осуществлено. Перемещение будет продолжаться до тех пор, пока сигнал разрешения (**<on>**) будет выставлен в 1 и пока не будет выставлен сигнал **<off>**. Перемещение осуществляется на рабочих ускорениях ($N6n00$).

Функция вернёт 1, если движение было окончено или отменено.



Если через AXIS_CONTROL было начато движение (<on> выставлен в 1), то движение по основной программе переопределяется, то есть движение по основной программе интерполируется, но фактически не осуществляется. После того, как AXIS_CONTROL завершит своё исполнение разница между текущим положением по программе и фактическим положением будет запомнена и будет применяться к текущим координатам по программе до тех пор, пока не будет найден кадр с перемещением по данной координате или пока не исполнится команда SYNC. SYNC, в свою очередь, передаст в интерпретатор фактическую координату.

Внимание! Функцию нельзя использовать в режиме выхода в 0, а также во время перехода из ручного в автоматический режим.

Пример. Выполнение нескольких последовательных перемещений по оси X

```
1 // Инициализация
2 // Ось X под номером 1:
3 #define X 1
4 CV10 = 6; // 6 разных поочерёдных перемещений
5 ...
6 // Быстрая секция ПЛК
7 // Сброс движения по кнопке сброса:
8 reset_C10 = C10 && (reset_C10 || SB_RESET);
9
10 CI10(0); // Эмулирование переднего фронта для непрерывного счёта
11 // Перемещение:
12 CI10(((C10 == 0) && I204_2) || // Начало движения по нажатию
      кнопки
13      (C10 == 1) && AXIS_CONTROL(1, reset_C10, X, POS_TO_RT
      (1), 100) ||
14      (C10 == 2) && AXIS_CONTROL(1, reset_C10, X, POS_TO_RT
      (-1), 100) ||
15      (C10 == 3) && AXIS_CONTROL(1, reset_C10, X, POS_TO_RT
      (2), 1000) ||
16      (C10 == 4) && AXIS_CONTROL(1, reset_C10, X, POS_TO_RT
      (-2), 1000) ||
17      (C10 == 5) && AXIS_CONTROL(1, reset_C10, X, POS_TO_RT
      (3), 1000) ||
18      (C10 == 6) && AXIS_CONTROL(1, reset_C10, X, POS_TO_RT
      (-3), 1000));
```



1.16.7 Функция привязки текущего инструмента от ПЛК

Синтаксис:

```
TOOL_MEASURE(<axis>, <val>);
```

ВАЖНО: привязка происходит для текущего инструмента (CH1_T_ID_W).

В аргументах функции указываем: номер оси (<axis>), по которой осуществляется привязка; новую координату инструмента (<val>).

Пример. Привязка текущего инструмента

```
1 TOOL_MEASURE(4, PU6); // Привязка произойдет по 4 оси со  
   значением привязки взятом из параметра PU6
```

1.16.8 Функция задания смещения от ПЛК

Синтаксис:

```
OFFSET_MEASURE(<axis>, <offset_id>, <val>);
```

В аргументах функции указываем: номер оси (<axis>), по которой хотим назначить смещение; номер привязки нуля детали (54 — для G54, 55 — для G55) — <offset_id>; значение привязки (<val>).

Пример. Функция задания смещения

```
1 OFFSET_MEASURE(1, 54, 25); // Задать в качестве нуля детали  
   координату 25 по 1 оси для системы смещения G54
```

1.16.9 Функция задания коррекции смещения от ПЛК

Синтаксис:

```
OFFSET_A_SET(<axis>, <offset_id>, <val>);
```

В аргументах функции указываем: номер оси (<axis>), по которой хотим назначить коррекцию смещения; номер привязки нуля детали (54 — для G54, 55 — для G55) — <offset_id>; значение привязки (<val>).

Пример. Функция задания коррекции смещения

```
1 OFFSET_A_SET(1, 54, 25); // Скорректировать смещение по 1 оси  
   для системы смещения G54 на 25
```



1.16.10 Функция увеличения коррекции смещения от ПЛК

Синтаксис:

```
OFFSET_A_INC(<axis>, <offset_id>, <val>);
```

В аргументах функции указываем: номер оси (**<axis>**), по которой хотим увеличить коррекцию смещения; номер привязки нуля детали (54 — для G54, 55 — для G55) — **<offset_id>**; значение, на которое хотим увеличить смещение (**<val>**).

Пример. Функция увеличения коррекции смещения

```
1 OFFSET_A_SET(1, 54, 5); // Увеличить коррекцию смещения по 1 оси  
   для системы смещения G54 на 5
```

1.16.11 Функция задания высоты текущего инструмента от ПЛК

Синтаксис:

```
TOOL_H_SET(<axis>, <val>);
```

В аргументах функции указываем: номер оси (**<axis>**), по которой откладывается высота текущего инструмента; значение высоты (**<val>**).

Пример. Функция задания высоты текущего инструмента

```
1 TOOL_H_SET(1, 4); // Отложить значение высоты инструмента,  
   равное 4, по 1 оси
```

1.16.12 Функция задания высоты корректора от ПЛК

Синтаксис:

```
CORR_H_SET(<id>, <new_val>);
```

В аргументах функции указываем: номер корректора (**<id>**), для которого откладывается высота; значение высоты (**<new_val>**).

Пример. Функция задания высоты текущего инструмента

```
1 CORR_H_SET(11, 4); // Задать для корректора H11 значение высоты,  
   равное 4
```



1.16.13 Функция получения номера слота

Синтаксис:

```
TOOL_GET_SLOT_N(<tool_n>, <binding_n>);
```

В аргументах функции указываем: номер инструмента (<**tool_n**>); номер привязки (<**binding_n**>).

Пример. Функция получения номера слота

```
1 TOOL_GET_SLOT_N(2, 1); // Получить номер слота, в котором  
находится инструмент под номером 2 с привязкой 1
```

1.16.14 Функция получения номера инструмента в слоте

Синтаксис:

```
SLOT_GET_TOOL_N(<slot_n>);
```

В аргументах функции указываем: номер слота (<**slot_n**>).

Пример. Функция получения номера инструмента в слоте

```
1 SLOT_GET_TOOL_N(1); // Получить номер инструмента, находящегося  
в 1-м слоте
```

1.16.15 Функция вставки инструмента в слот

Синтаксис:

```
TOOL_SET_SLOT_N(<tool_n>, <binding_n>, <slot_n>);
```

В аргументах функции указываем: номер инструмента (<**tool_n**>); номер привязки (<**binding_n**>); номер слота (<**slot_n**>).

Пример. Функция вставки инструмента в слот

```
1 TOOL_SET_SLOT_N(3, 1, 2); // Поместить инструмент под номером 3  
с 1-й привязкой во 2-й слот
```



1.16.16 Функция получения количества магазинов инструментов

Синтаксис:

```
TURRETS_COUNT();
```

Пример. Функция получения количества магазинов инструментов

```
1 TURRETS_COUNT(); // Получить количество магазинов инструментов в  
   системе
```

1.16.17 Функция получения кода ошибки магазина инструментов

Синтаксис:

```
TURRET_ERRCODE(<turret_id>);
```

В аргументах функции указываем: номер магазина инструментов (<turret_id>).

Пример. Функция получения кода ошибки магазина инструментов

```
1 TURRET_ERRCODE(1); // Получить код ошибки 1-го магазина  
   инструментов
```

1.16.18 Функция получения типа магазина инструментов

Синтаксис:

```
TURRET_TYPE(<turret_id>);
```

В аргументах функции указываем: номер магазина инструментов (<turret_id>).

Пример. Функция получения кода ошибки магазина инструментов

```
1 TURRET_TYPE(1); // Получить тип 1-го магазина инструментов
```



1.16.19 Функция получения количества слотов в магазине инструментов

Синтаксис:

```
TURRET_SLOTS_COUNT(<turret_id>);
```

В аргументах функции указываем: номер магазина инструментов (<turret_id>).

Пример. Функция получения количества слотов в магазине инструментов

```
1 TURRET_SLOTS_COUNT(1); // Получить количество слотов в 1-м
   магазине инструментов
```

1.16.20 Функция получения номера слота в магазине инструментов по его идентификатору

Синтаксис:

```
TURRET_GET_SLOT_N(<turret_id>, <slot_id>);
```

В аргументах функции указываем: номер магазина инструментов (<turret_id>); идентификатор слота (<slot_id>).

Пример. Функция получения номера слота в магазине по его идентификатору

```
1 TURRET_GET_SLOT_N(1, 4); // Получить номер слота с
   идентификатором 4 в 1-м магазине инструментов
```

1.16.21 Функция получения идентификатора слота в магазине инструментов по его номеру

Синтаксис:

```
TURRET_GET_SLOT_ID(<turret_id>, <slot_n>);
```

В аргументах функции указываем: номер магазина инструментов (<turret_id>); номер слота (<slot_n>).

Пример. Функция получения идентификатора слота в магазине по его номеру

```
1 TURRET_GET_SLOT_ID(1, 3); // Получить идентификатор 3-го слота в
   1-м магазине инструментов
```



1.16.22 Функция получения идентификатора инструмента по номеру слота в магазине инструментов

Синтаксис:

```
SLOT_GET_TOOL_ID(<turret_id>, <slot_id>);
```

В аргументах функции указываем: номер магазина инструментов (<turret_id>); номер слота (<slot_id>).

Пример. Функция получения идентификатора инструмента по номеру слота в магазине инструментов

```
1 SLOT_GET_TOOL_ID(1, 3); // Получить идентификатор инструмента,  
находящегося в 3-м слоте 1-го магазина инструментов
```

1.16.23 Функция получения координат инструмента по номеру слота в магазине инструментов

Синтаксис:

```
SLOT_GET_AXIS_POS(<turret_id>, <slot_id>, <axis_id>);
```

В аргументах функции указываем: номер магазина инструментов (<turret_id>); номер слота (<slot_id>); номер оси (<axis_id>).

Пример. Функция получения координат инструмента по номеру слота в магазине инструментов

```
1 SLOT_GET_AXIS_POS(2, 3, 1); // Получить координату по 1-й оси  
инструмента, находящегося в 3-м слоте 2-го магазина  
инструментов
```

1.16.24 Функции получения/записи битов переменной

Для получения/записи отдельных битов переменной в программе электроавтоматики используют следующие функции:

- **GETBIT(<пер-ая>, <ном_бита>)** — для получения соответствующего бита из переменной;
- **SETBIT(<пер-ая>, <ном_бита>, <значение>)** — для записи отдельного бита в переменную.



Пример. Работа с битовыми переменными

```
1 M100_1 = GETBIT(P16, 1); // Получение первого бита из параметра  
пользователя P16 (N8516 – в параметрах) и запись этого  
значения в M-ячейку  
2 if (GETBIT(M99, 5)) { SETBIT(M99,1,0); } // Если 5-й бит ячейки  
M99 будет равен 1, то первый бит этой M-ячейки установится в  
0
```

1.16.25 Функции режима обучения

Первым аргументом **<enable>** во все функции передается разрешение или запрет на ее выполнение (1 или 0 соответственно).

Функция включения режима обучения в графическом интерфейсе

Синтаксис:

```
GUI_SCAN_ON(<enable>);
```

Функция выполняет все операции, необходимые для записи кадров в режиме обучения, используя графический интерфейс.

Функции включения режима обучения от ПЛК

Синтаксис:

```
SCAN_ON(<enable>, <mode>);
```

В аргументах функции указываем: разрешение на выполнение (**<enable>**) и режим работы (**<mode>**). Реализованы следующие режимы работы:

- SCANNER_MODE_AUTO — режим автоматического сканирования;
- SCANNER_MODE_MANUAL — режим ручного обучения.

Функция SCAN_ON самостоятельно записывает ячейку CH1_SCAN_SET_ON_IO = 1, а также формирует кадр с быстрым переходом в текущую позицию (G0) по всем осям текущего канала.



Функции записи координат в режиме обучения

Синтаксис:

```
// запись координат всех осей ЧПУ;  
SCAN_AXES_POS(<enable>);  
// запись координат всех осей текущего канала;  
SCAN_AXIS_POS_BY_CH(<enable>);  
// запись координаты оси с указанным номером;  
SCAN_AXIS_POS(<enable>, <axis_num>);
```

Функции записи команд

Синтаксис:

```
SCAN_G(<enable>, <code>, <mod>);
```

Данная функция сформирует G-код с номером **<code>** и модификатором **<mod>**. Функция **SCAN_G** самостоятельно записывает ячейки **CH1_SCAN_G_OI = 1**, **CH1_SCAN_G_CODE_W = <code>** и **CH1_SCAN_G_MOD_W = <mod>**.

Синтаксис:

```
SCAN_G0(<enable>);
```

Данная функция сформирует G-код с номером = 0 и модификатором = 0. Функция **SCAN_G0** самостоятельно записывает ячейки **CH1_SCAN_G_OI = 1**, **CH1_SCAN_G_CODE_W = 0** и **CH1_SCAN_G_MOD_W = 0**.

Синтаксис:

```
SCAN_G1(<enable>);
```

Данная функция сформирует G-код с номером = 1 и модификатором = 0. Функция **SCAN_G1** самостоятельно записывает ячейки **CH1_SCAN_G_OI = 1**, **CH1_SCAN_G_CODE_W = 1** и **CH1_SCAN_G_MOD_W = 0**.

Синтаксис:

```
SCAN_M(<enable>, <code>, <mod>);
```

Данная функция сформирует M-команду с номером **<code>** и модификатором **<mod>**. Функция **SCAN_M** самостоятельно записывает ячейки **CH1_SCAN_M_OI = 1**, **CH1_SCAN_M_CODE_W = <code>** и **CH1_SCAN_M_MOD_W = <mod>**.

Синтаксис:



```
SCAN_F(<enable>, <speed>);
```

Данная функция сформирует кадр задания подачи F<**speed**>. Функция SCAN_F самостоятельно записывает ячейки CH1_SCAN_F_OI = 1 и CH1_SCAN_F_VAL_W = <speed>.

Синтаксис:

```
SCAN_T(<enable>, <num>, <binding_num>);
```

Данная функция сформирует T-команду для выбора инструмента с номером = <**num**> и номером привязки = <**binding_num**>. Функция SCAN_T самостоятельно записывает ячейки CH1_SCAN_T_OI = 1, CH1_SCAN_T_N_W = <num> и CH1_SCAN_T_B_W = <binding_num>.

1.16.26 Функции создания и работы с пультом быстрых программ

Пульт быстрых программ может быть разделен на несколько профилей. Для корректного отображения пульта необходимо создать хотя бы один профиль.

Функция создания профиля в пульте быстрых программ

Синтаксис:

```
FAST_PRG_NEW_PROFILE(<name>);
```

В аргументах функции указываем: название профиля (<**name**>), которое будет отображено как метка на пульте.

Именно на последний созданный профиль будут добавляться все кнопки, созданные впоследствии.

Функции добавления кнопки на пульт быстрых программ

Синтаксис:

```
FAST_PRG_ADD(<x>, <y>, <width>, <height>,  
             <title>, <icon_name>, <flags>);
```

В аргументах функции указываем: номер столбца, в которой необходимо поместить кнопку (<**x**>); номер строки, в которую необходимо поместить кнопку (<**y**>); ширина кнопки как количество занимаемых ей столбцов (<**width**>); высота кнопки как количество занимаемых ей строк (<**height**>); надпись на



кнопке (**<title>**); название иконки на кнопке (**<icon_name>**); флаг, который определяет специальные возможности (**<flags>**).

В качестве последнего аргумента **<flags>** может быть передано значение DEFAULTS (по умолчанию) либо AUTOCLOSE (кнопка автоматически закрывает пульт по ее нажатию).

Функции возвращают идентификатор добавленной кнопки.

```
FAST_PRG_ADD_PRG(<x>, <y>, <width>, <height>,  
                 <title>, <icon_name>, <flags>,  
                 <prg_name>);
```

Аргумент **<prg_name>** определяет название программы, которая будет запущена на исполнение по нажатию кнопки. Файл, содержащий код программы, должен иметь расширение .nc и находиться в директории /home/titanium/fast_prg.

Также этот аргумент может быть задан отдельно с помощью специальной функции.

Функция задания управляющей программы на кнопку пульта быстрых программ

Синтаксис:

```
FAST_PRG_BUTTON_ADD_PRG(<button_id>, <prg_name>);
```

В аргументах функции указываем: идентификатор кнопки (**<button_id>**); название файла управляющей программы (**<prg_name>**).

Функция задания номера канала, в котором будет обрабатываться нажатие кнопки пульта быстрых программ

Синтаксис:

```
FAST_PRG_BUTTON_SET_CHANNEL(<button_id>, <channel_num>);
```

В аргументах функции указываем: идентификатор кнопки (**<button_id>**); номер канала (**<channel_num>**).

Функция создания пульта быстрых программ

Синтаксис:

```
FAST_PRG_COMMIT();
```

Данная функция подтверждает, что все необходимые виджеты пульта созданы и пульт доступен для отображения и работы. При отсутствии этой команды кнопка вызова пульта быстрых программ на главном окне останется заблокированной.



Функция нажатия кнопки пульта быстрых программ

Синтаксис:

```
FAST_PRG_CLICK(<enable>, <button_id>);
```

В аргументах функции указываем: разрешение или запрет на активацию (**<enable>**); идентификатор кнопки (**<button_id>**).

Система параметров устройства ЧПУ

Для установки текущей конфигурации устройства ЧПУ, версии программного обеспечения, особенностей станка и условий его эксплуатации служит система параметров. Все параметры делятся на несколько групп в зависимости от назначения и уровня доступа. В общем виде систему параметров можно представить следующим образом.

- системные параметры (1);
- общие базовые станочные параметры (2);
- общие станочные параметры (3);
- общие технологические параметры (4);
- общие наладочные параметры (5);
- базовые станочные параметры оси (2);
- станочные параметры оси (3);
- наладочные параметры оси (4);
- группа параметров пользователя (5);
- базовые станочные параметры маховиков (2).

В скобках указаны уровни доступа к параметрам:

- 1 — параметры доступны изготовителям УЧПУ;
- 2 — параметры доступны изготовителям станков;
- 3 — параметры доступны наладчикам станков;
- 4 — параметры доступны технологам, эксплуатирующим станки;
- 5 — параметры доступны операторам станков.

Параметры просматриваются и вводятся отдельными группами. При этом на индикацию выводятся название параметра, его размерность и внутреннее системное наименование, которое указывается только для информации.



Ниже приведен полный список параметров, где в конце описания каждого параметра указан диапазон его изменения.

2.1 Системные параметры

- N1.** Период работы системного таймера.
- N2.** Выделение отдельного ядра процессора под задачи реального времени (только для многоядерных процессоров): «0» — отключено; «1» — включено; «2» — включено с изоляцией от системных процессов ОС. Включение параметра сильно снижает производительность системы.
- N3.** Разрешить запись данных на основной накопитель информации: «0» — запретить; «1» — разрешить.
- N5.** Уровень отладки: «0» — не показывать системные сообщения; «1» — показывать диагностические сообщения; «2» — показывать отладочные сообщения.
- N6.** Максимальная частота сообщений одного типа от процессов реального времени. Если частота сообщений одного типа превышает заданную, то сообщения группируются в одно, где сообщается сколько раз сообщение было отправлено. Работает совместно с **N7**.
- N7.** Период времени, для которого задаётся частота из параметра **N6**.
- N10.** Для драйвера `snc11` — базовый адрес порта первого блока ЦАП в шестнадцатеричном виде (230 — для модели ЧПУ Titanium; 2C0 — для модели ЧПУ CNC8). Для драйвера `ethercat` — параметр не используется. Для драйвера `titan` — имя устройства, которым оно представлено в файловой системе (например, `ttyUSB0`) или **auto** для автоматического определения названия подходящего устройства. По умолчанию: `auto`.
- N11.** ID модуля, к которому подключается драйвер ЦАП.
- N12.** Имя драйвера, обрабатывающего первый блок каналов ЦАП (например: `snc11`, `ethercat`, `titan`).
- N13.** Количество каналов блока ЦАП. Для драйвера `snc11` — 4 канала, независимо от количества задействованных из них. Для драйвера `titan` — 6 каналов, независимо от количества задействованных из них.
- N20.** Для драйвера `snc11` — базовый адрес порта второго блока ЦАП в шестнадцатеричном виде (240 — для модели ЧПУ Titanium; 2B0 — для модели



ЧПУ CNC8). Для драйвера ethercat — параметр не используется. Для драйвера titan — имя устройства, которым оно представлено в файловой системе (например, ttyUSB0).

- N21.** ID модуля, к которому подключается драйвер ЦАП.
- N22.** Имя драйвера, обрабатывающего второй блок каналов ЦАП (например: cnc11, ethercat, titan).
- N23.** Количество каналов блока ЦАП. Для драйвера cnc11 — 4 канала, независимо от количества задействованных из них. Для драйвера titan — 6 каналов, независимо от количества задействованных из них.
- N30.** Для драйвера cnc11 — базовый адрес порта второго блока ЦАП в шестнадцатеричном виде (250 — для модели ЧПУ Titanium; 2E0 — для модели ЧПУ CNC8). Для драйвера ethercat — параметр не используется. Для драйвера titan — имя устройства, которым оно представлено в файловой системе (например, ttyUSB0).
- N31.** ID модуля, к которому подключается драйвер ЦАП.
- N32.** Имя драйвера, обрабатывающего третий блок каналов ЦАП (например: cnc11, ethercat, titan).
- N33.** Количество каналов блока ЦАП. Для драйвера cnc11 — 4 канала, независимо от количества задействованных из них. Для драйвера titan — 6 каналов, независимо от количества задействованных из них.
- N40.** Для драйвера cnc11 — базовый адрес порта второго блока ЦАП в шестнадцатеричном виде (260 — для модели ЧПУ Titanium; 2F0 — для модели ЧПУ CNC8). Для драйвера ethercat — параметр не используется. Для драйвера titan — имя устройства, которым оно представлено в файловой системе (например, ttyUSB0).
- N41.** ID модуля, к которому подключается драйвер ЦАП.
- N42.** Имя драйвера, обрабатывающего четвертый блок каналов ЦАП (например: cnc11, ethercat, titan).
- N43.** Количество каналов блока ЦАП. Для драйвера cnc11 — 4 канала, независимо от количества задействованных из них. Для драйвера titan — 6 каналов, независимо от количества задействованных из них.
- N50.** Для драйвера cnc11 — базовый адрес порта первого блока оцифровки в шестнадцатеричном виде (200 — для модели ЧПУ Titanium; 2F0 — для



модели ЧПУ CNC8). Для драйвера ethercat — параметр не используется. Для драйвера titan — имя устройства, которым оно представлено в файловой системе (например, ttyUSB0) или **auto** для автоматического определения названия подходящего устройства. По умолчанию: auto.

- N51.** ID модуля, к которому подключается драйвер блока оцифровки.
- N52.** Имя драйвера, обрабатывающего первый блок оцифровки каналов ЦАП (например: cnc11, ethercat, titan).
- N53.** Количество каналов блока оцифровки. Для драйвера cnc11 — 3 канала, независимо от количества задействованных из них. Для драйвера titan — 6 каналов, независимо от количества задействованных из них.
- N54.** Разрядность блока оцифровки 24-CNC11.
- N60.** Для драйвера cnc11 — базовый адрес порта второго блока оцифровки в шестнадцатеричном виде (210 — для модели ЧПУ Titanium; 2F0 — для модели ЧПУ CNC8). Для драйвера ethercat — параметр не используется. Для драйвера titan — имя устройства, которым оно представлено в файловой системе (например, ttyUSB0).
- N61.** ID модуля, к которому подключается драйвер блока оцифровки.
- N62.** Имя драйвера, обрабатывающего второй блок оцифровки каналов ЦАП (например: cnc11, ethercat, titan).
- N63.** Количество каналов блока оцифровки. Для драйвера cnc11 — 3 канала, независимо от количества задействованных из них. Для драйвера titan — 6 каналов, независимо от количества задействованных из них.
- N64.** Разрядность блока оцифровки 24-CNC11.
- N70.** Для драйвера cnc11 — базовый адрес порта третьего блока оцифровки в шестнадцатеричном виде (220 — для модели ЧПУ Titanium; 2F0 — для модели ЧПУ CNC8). Для драйвера ethercat — параметр не используется. Для драйвера titan — имя устройства, которым оно представлено в файловой системе (например, ttyUSB0).
- N71.** ID модуля, к которому подключается драйвер блока оцифровки.
- N72.** Имя драйвера, обрабатывающего третий блок оцифровки каналов ЦАП (например: cnc11, ethercat, titan).
- N73.** Количество каналов блока оцифровки. Для драйвера cnc11 — 3 канала, независимо от количества задействованных из них. Для драйвера titan — 6 каналов, независимо от количества задействованных из них.



- N74.** Разрядность блока оцифровки 24-CNC11.
- N80.** Для драйвера `snc11` — базовый адрес порта четвертого блока оцифровки в шестнадцатеричном виде (230 — для модели ЧПУ Titanium; 2F0 — для модели ЧПУ CNC8). Для драйвера `ethercat` — параметр не используется. Для драйвера `titan` — имя устройства, которым оно представлено в файловой системе (например, `ttyUSB0`).
- N81.** ID модуля, к которому подключается драйвер блока оцифровки.
- N82.** Имя драйвера, обрабатывающего четвертый блок оцифровки каналов ЦАП (например: `snc11`, `ethercat`, `titan`).
- N83.** Количество каналов блока оцифровки. Для драйвера `snc11` — 3 канала, независимо от количества задействованных из них. Для драйвера `titan` — 6 каналов, независимо от количества задействованных из них.
- N84.** Разрядность блока оцифровки 24-CNC11.
- N101.** Размер буфера кадров интерпретатора УП, по умолчанию: 1000.
- N102.** Каталог размещения файлов программ, по умолчанию: «`rogram`».
- N107.** Имя каталога с файлами электроавтоматики, рекомендуется делать по названию станка, например: «2с150».
- N108.** Частота запуска сверхбыстрой секции ПЛК, мс. Например, «10» будет означать 10 мс. Если сверхбыстрая секция не используется, то данное значение должно быть равно значению параметра **N109**. Значение данного параметра должно быть кратным частоте запуска секции реального времени (**N1**).
- N109.** Частота запуска быстрой секции ПЛК, мс. Например, «10» будет означать 10 мс. Если быстрая и сверхбыстрая секции не используются, то данное значение должно быть равно значению параметра **N110**. Значение данного параметра должно быть кратным частоте запуска сверхбыстрой секции (**N108**), в противном случае значение будет округлено до ближайшего кратного.
- N110.** Частота запуска медленной секции ПЛК. Например, «100» будет означать 100 мс. Значение данного параметра должно быть кратным частоте запуска сверхбыстрой секции (**N108**), в противном случае значение будет округлено до ближайшего кратного.
- N200.** Начальный номер байта 1-й группы входов, 1..Max1п.



- N201.** Количество байт в 1-й группе входов.
- N202.** Для драйвера sys — базовый адрес 1-й группы входов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N203.** ID подключаемого драйвера для 1-й группы входов.
- N204.** Тип логики 1-й группы входов: «0» — прямая; «1» — инверсная.
- N205.** Начальный номер байта 2-й группы входов, 1..Max1п.
- N206.** Количество байт в 2-й группе входов.
- N207.** Для драйвера sys — базовый адрес 2-й группы входов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N208.** ID подключаемого драйвера для 2-й группы входов.
- N209.** Тип логики 2-й группы входов: «0» — прямая; «1» — инверсная.
- N210.** Начальный номер байта 3-й группы входов, 1..Max1п.
- N211.** Количество байт в 3-й группе входов.
- N212.** Для драйвера sys — базовый адрес 3-й группы входов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N213.** ID подключаемого драйвера для 3-й группы входов.
- N214.** Тип логики 3-й группы входов: «0» — прямая; «1» — инверсная.
- N215.** Начальный номер байта 4-й группы входов, 1..Max1п.
- N216.** Количество байт в 4-й группе входов.
- N217.** Для драйвера sys — базовый адрес 4-й группы входов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N218.** ID подключаемого драйвера для 4-й группы входов.



- N219.** Тип логики 4-й группы входов: «0» — прямая; «1» — инверсная.
- N220.** Начальный номер байта 5-й группы входов, 1..Max1п.
- N221.** Количество байт в 5-й группе входов.
- N222.** Для драйвера sys — базовый адрес 5-й группы входов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N223.** ID подключаемого драйвера для 5-й группы входов.
- N224.** Тип логики 5-й группы входов: «0» — прямая; «1» — инверсная.
- N225.** Начальный номер байта 6-й группы входов, 1..Max1п.
- N226.** Количество байт в 6-й группе входов.
- N227.** Для драйвера sys — базовый адрес 6-й группы входов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N228.** ID подключаемого драйвера для 6-й группы входов.
- N229.** Тип логики 6-й группы входов: «0» — прямая; «1» — инверсная.
- N230.** Начальный номер байта 7-й группы входов, 1..Max1п.
- N231.** Количество байт в 7-й группе входов.
- N232.** Для драйвера sys — базовый адрес 7-й группы входов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N233.** ID подключаемого драйвера для 7-й группы входов.
- N234.** Тип логики 7-й группы входов: «0» — прямая; «1» — инверсная.
- N235.** Начальный номер байта 8-й группы входов, 1..Max1п.
- N236.** Количество байт в 8-й группе входов.
- N237.** Для драйвера sys — базовый адрес 8-й группы входов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.



- N238.** ID подключаемого драйвера для 8-й группы входов.
- N239.** Тип логики 8-й группы входов: «0» — прямая; «1» — инверсная.
- N240.** Начальный номер байта 9-й группы входов, 1..Max1п.
- N241.** Количество байт в 9-й группе входов.
- N242.** Для драйвера sys — базовый адрес 9-й группы входов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N243.** ID подключаемого драйвера для 9-й группы входов.
- N244.** Тип логики 9-й группы входов: «0» — прямая; «1» — инверсная.
- N245.** Начальный номер байта 10-й группы входов, 1..Max1п.
- N246.** Количество байт в 10-й группе входов.
- N247.** Для драйвера sys — базовый адрес 10-й группы входов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N248.** ID подключаемого драйвера для 10-й группы входов.
- N249.** Тип логики 10-й группы входов: «0» — прямая; «1» — инверсная.
- N300.** Начальный номер байта 1-й группы выходов, 1..MaxOut.
- N301.** Количество байт в 1-й группе выходов.
- N302.** Для драйвера sys — базовый адрес 1-й группы выходов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N303.** ID подключаемого драйвера для 1-й группы выходов.
- N304.** Тип логики 1-й группы выходов: «0» — прямая; «1» — инверсная.
- N305.** Начальный номер байта 2-й группы выходов, 1..MaxOut.
- N306.** Количество байт в 2-й группе выходов.



- N307.** Для драйвера sys — базовый адрес 2-й группы выходов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N308.** ID подключаемого драйвера для 2-й группы выходов.
- N309.** Тип логики 2-й группы выходов: «0» — прямая; «1» — инверсная.
- N310.** Начальный номер байта 3-й группы выходов, 1..MaxOut.
- N311.** Количество байт в 3-й группе выходов.
- N312.** Для драйвера sys — базовый адрес 3-й группы выходов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N313.** ID подключаемого драйвера для 3-й группы выходов.
- N314.** Тип логики 3-й группы выходов: «0» — прямая; «1» — инверсная.
- N315.** Начальный номер байта 4-й группы выходов, 1..MaxOut.
- N316.** Количество байт в 4-й группе выходов.
- N317.** Для драйвера sys — базовый адрес 4-й группы выходов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N318.** ID подключаемого драйвера для 4-й группы выходов.
- N319.** Тип логики 4-й группы выходов: «0» — прямая; «1» — инверсная.
- N320.** Начальный номер байта 5-й группы выходов, 1..MaxOut.
- N321.** Количество байт в 5-й группе выходов.
- N322.** Для драйвера sys — базовый адрес 5-й группы выходов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N323.** ID подключаемого драйвера для 5-й группы выходов.
- N324.** Тип логики 5-й группы выходов: «0» — прямая; «1» — инверсная.



- N325.** Начальный номер байта 6-й группы выходов, 1..MaxOut.
- N326.** Количество байт в 6-й группе выходов.
- N327.** Для драйвера sys — базовый адрес 6-й группы выходов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N328.** ID подключаемого драйвера для 6-й группы выходов.
- N329.** Тип логики 6-й группы выходов: «0» — прямая; «1» — инверсная.
- N330.** Начальный номер байта 7-й группы выходов, 1..MaxOut.
- N331.** Количество байт в 7-й группе выходов.
- N332.** Для драйвера sys — базовый адрес 7-й группы выходов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N333.** ID подключаемого драйвера для 7-й группы выходов.
- N334.** Тип логики 7-й группы выходов: «0» — прямая; «1» — инверсная.
- N335.** Начальный номер байта 8-й группы выходов, 1..MaxOut.
- N336.** Количество байт в 8-й группе выходов.
- N337.** Для драйвера sys — базовый адрес 8-й группы выходов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N338.** ID подключаемого драйвера для 8-й группы выходов.
- N339.** Тип логики 8-й группы выходов: «0» — прямая; «1» — инверсная.
- N340.** Начальный номер байта 9-й группы выходов, 1..MaxOut.
- N341.** Количество байт в 9-й группе выходов.
- N342.** Для драйвера sys — базовый адрес 9-й группы выходов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.



- N343.** ID подключаемого драйвера для 9-й группы выходов.
- N344.** Тип логики 9-й группы выходов: «0» — прямая; «1» — инверсная.
- N345.** Начальный номер байта 10-й группы выходов, 1..MaxOut.
- N346.** Количество байт в 10-й группе выходов.
- N347.** Для драйвера sys — базовый адрес 10-й группы выходов в шестнадцатеричном виде. Для драйвера ethercat — параметр не используется. Для драйвера titan — сквозное смещение в байтах относительно первой подключенной платы.
- N348.** ID подключаемого драйвера для 10-й группы выходов.
- N349.** Тип логики 10-й группы выходов: «0» — прямая; «1» — инверсная.
- N410.** Имя 1-го подключаемого модуля.
- N411.** Файл конфигурации 1-го модуля.
- N412.** Номер мастера, используемого в 1-ом модуле (Ethercat).
- N413.** Имя 2-го подключаемого модуля.
- N414.** Файл конфигурации 2-го модуля.
- N415.** Номер мастера, используемого в 2-ом модуле (Ethercat).
- N416.** Имя 3-го подключаемого модуля.
- N417.** Файл конфигурации 3-го модуля.
- N418.** Номер мастера, используемого в 3-ом модуле (Ethercat).
- N420.** ID модуля, подключаемого к 1-му драйверу входов/выходов.
- N421.** Имя 1-го драйвера входов/выходов.
- N422.** ID модуля, подключаемого к 2-му драйверу входов/выходов.
- N423.** Имя 2-го драйвера входов/выходов.
- N424.** ID модуля, подключаемого к 3-му драйверу входов/выходов.
- N425.** Имя 3-го драйвера входов/выходов.
- N426.** ID модуля, подключаемого к 4-му драйверу входов/выходов.
- N427.** Имя 4-го драйвера входов/выходов.



- N428.** ID модуля, подключаемого к 5-му драйверу входов/выходов.
- N429.** Имя 5-го драйвера входов/выходов.
- N430.** Имя конфигурации 1-й группы входов/выходов.
- N431.** Имя конфигурации 2-й группы входов/выходов.
- N432.** Имя конфигурации 3-й группы входов/выходов.
- N433.** Имя конфигурации 4-й группы входов/выходов.
- N434.** Имя конфигурации 5-й группы входов/выходов.
- N435.** Имя конфигурации 6-й группы входов/выходов.
- N436.** Имя конфигурации 7-й группы входов/выходов.
- N437.** Имя конфигурации 8-й группы входов/выходов.
- N438.** Имя конфигурации 9-й группы входов/выходов.
- N439.** Имя конфигурации 10-й группы входов/выходов.
- N440.** Имя устройства, которым оно представлено в файловой системе (например, ttyUSB0) 1-й группы входов/выходов или **auto** для автоопределения. По умолчанию: auto.
- N441.** Имя устройства, которым оно представлено в файловой системе (например, ttyUSB0) 2-й группы входов/выходов или **auto** для автоопределения. По умолчанию: auto.
- N442.** Имя устройства, которым оно представлено в файловой системе (например, ttyUSB0) 3-й группы входов/выходов или **auto** для автоопределения. По умолчанию: auto.
- N443.** Имя устройства, которым оно представлено в файловой системе (например, ttyUSB0) 4-й группы входов/выходов или **auto** для автоопределения. По умолчанию: auto.
- N444.** Имя устройства, которым оно представлено в файловой системе (например, ttyUSB0) 5-й группы входов/выходов или **auto** для автоопределения. По умолчанию: auto.
- N445.** Имя устройства, которым оно представлено в файловой системе (например, ttyUSB0) 6-й группы входов/выходов или **auto** для автоопределения. По умолчанию: auto.



- N446.** Имя устройства, которым оно представлено в файловой системе (например, ttyUSB0) 7-й группы входов/выходов или **auto** для автоопределения. По умолчанию: auto.
- N447.** Имя устройства, которым оно представлено в файловой системе (например, ttyUSB0) 8-й группы входов/выходов или **auto** для автоопределения. По умолчанию: auto.
- N448.** Имя устройства, которым оно представлено в файловой системе (например, ttyUSB0) 9-й группы входов/выходов или **auto** для автоопределения. По умолчанию: auto.
- N449.** Имя устройства, которым оно представлено в файловой системе (например, ttyUSB0) 10-й группы входов/выходов или **auto** для автоопределения. По умолчанию: auto.
- N501.** Разрешить запись сообщений с ошибками на диск: «0» — запретить; «1» — разрешить. Работает совместно с параметром **N502**.
- N502.** Размер одного файла журнала, 1М...15М.
- N503.** Количество файлов журнала в архиве, 5...20.
- N505.** Включить сглаживание линий при отображении траектории в визуализации.
- N506.** Включить слежение за траекторией движения инструмента в визуализации.
- N507.** Зарезервировано.
- N508.** Интервал времени между считываниями текущих координат, 0...999999 мс. По умолчанию: 100000.
- N509.** Ограничение памяти, выделяемой под хранение траектории движения инструмента. По умолчанию: 32М (т. е. 32 МиБ).
- N510.** Зарезервировано.
- N511.** Ограничение памяти, выделяемой под хранение траектории предварительного просмотра управляющей программы. По умолчанию: 32М (т. е. 32 МиБ).
- N512.** Минимальное расстояние между точками визуализации движения инструмента. Увеличение расстояния уменьшает точность изображения траектории, но увеличивает производительность при отрисовке и уменьшает потребление памяти.



- N513.** Интервал времени между считываниями очередного кадра интерпретатора в предварительном просмотре, 0...999999 мс. По умолчанию: 100000.
- N514.** Интервал времени между перерисовкой графики, 0...999999 мс. По умолчанию: 100000.
- N520.** Минимальный объём свободной памяти ЧПУ, при котором разрешено исполнение управляющих программ. По умолчанию: 10М (т. е. 10 МиБ).
- N521.** Включить поддержку облегчённого интерфейса главного окна системы ЧПУ: «0» — выключить, «1» — включить. По умолчанию: «0». Параметр устарел и будет удалён в одной из следующих версий ЧПУ.
- N522.** Название темы оформления графического интерфейса системы ЧПУ. По умолчанию: «titanium-theme».
- N523.** Название окна, отображаемого по умолчанию в качестве главного окна: «main», «light» (облегчённый интерфейс), «view» (двухканальная версия главного окна). По умолчанию: «main».
- N524.** Частота обновления данных на экране, 1 Гц...60 Гц. При указании значения «0» происходит непрерывное обновление экрана. Рекомендуемый диапазон: 10-12. По умолчанию: 12.
- N525.** Показывать окно предварительного просмотра графики в диалоге открытия файла: «0» — не показывать; «1» — показывать.
- N526.** Показывать начало текста программы в диалоге открытия файла: «0» — не показывать; «1» — показывать.
- N530.** Отображать в главном окне величину остатка, который необходимо проехать до конца текущего кадра: «0» — не отображать; «1» — отображать. По умолчанию: «1».
- N531.** Отображать в главном окне скорость движения по оси: «0» — не отображать; «1» — отображать. По умолчанию: «0».
- N532.** Количество строк в программируемой через ПЛК клавиатуре (1...15). По умолчанию: 6.
- N533.** Количество столбцов в программируемой через ПЛК клавиатуре (1...20). По умолчанию: 12.
- N536.** Поменять местами расположение каналов в окне отображения информации по двух каналам.



- N537.** Прозрачность всплывающего виртуального пульта оператора при нажатии на кнопку пульта.
- N538.** Прозрачность всплывающего виртуального пульта оператора, 0.0...1.0.
- N539.** Разрешить изменение процентовок из главного окна.
- N541.** Максимальное количество сообщений, показываемых в диалоговом окне, вызываемом из модуля ПЛК. По умолчанию: 10.
- N600.** Пульт №1 включен: «0» — выключен; «1» — включен .
- N601.** Имя драйвера, обрабатывающего данные пульта №1.
- N602.** Имя пульта №1, которым он представлен в файловой системе (например, /dev/ttyUSB1) или **auto** — для автоопределения. По умолчанию: auto. Для драйвера хс параметр не используется.
- N603.** Скорость обмена данных для пульта №1. Для драйвера хс параметр не используется.
- N604.** Начальный адрес байта входов пульта №1.
- N605.** Начальный адрес байта выходов пульта №1.
- N606.** Тип управления пультом №1: «0» — пульт управляется драйвером; «1» — пульт управляется PLC.
- N607.** Имя файла конфигурации для пульта №1.
- N608.** Идентификатор модуля, к которому подключается драйвер пульта №1. Для драйвера spc11 — модуль sys, для драйвера хс — модуль usb.
- N609.** Тип физического подключения пульта №1. Для подключения через СОМ порт — com, для подключения через USB порт — usb. Для драйвера хс параметр не используется.
- N610.** Пульт №2 включен: «0» — выключен; «1» — включен.
- N611.** Имя драйвера, обрабатывающего данные пульта №2.
- N612.** Имя пульта №2, которым он представлен в файловой системе (например, /dev/ttyS0) или **auto** — для автоопределения. По умолчанию: auto.. Для драйвера хс параметр не используется.
- N613.** Скорость обмена данных для пульта №2. Для драйвера хс параметр не используется.



- N614.** Начальный адрес байта входов пульта №2.
- N615.** Начальный адрес байта выходов пульта №2.
- N616.** Тип управления пультом №2: «0» — пульт управляется драйвером; «1» — пульт управляется PLC.
- N617.** Имя файла конфигурации для пульта №2.
- N618.** Идентификатор модуля, к которому подключается драйвер пульта №2. Для драйвера snc11 — модуль sys, для драйвера хhc — модуль usb.
- N619.** Тип физического подключения пульта №2. Для подключения через СОМ порт — com, для подключения через USB порт — usb. Для драйвера хhc параметр не используется.
- N620.** Пульт №3 включен: «0» — выключен; «1» — включен.
- N621.** Имя драйвера, обрабатывающего данные пульта №3.
- N622.** Имя пульта №3, которым он представлен в файловой системе (например, /dev/ttyS0) или **auto** — для автоопределения. По умолчанию: auto.. Для драйвера хhc параметр не используется.
- N623.** Скорость обмена данных для пульта №3. Для драйвера хhc параметр не используется.
- N624.** Начальный адрес байта входов пульта №3.
- N625.** Начальный адрес байта выходов пульта №3.
- N626.** Тип управления пультом №3: «0» — пульт управляется драйвером; «1» — пульт управляется PLC.
- N627.** Имя файла конфигурации для пульта №3.
- N628.** Идентификатор модуля, к которому подключается драйвер пульта №3. Для драйвера snc11 — модуль sys, для драйвера хhc — модуль usb.
- N629.** Тип физического подключения пульта №3. Для подключения через СОМ порт — com, для подключения через USB порт — usb. Для драйвера хhc параметр не используется.
- N630.** Пульт №4 включен: «0» — выключен; «1» — включен.
- N631.** Имя драйвера, обрабатывающего данные пульта №4.



- N632.** Имя пульта №4, которым он представлен в файловой системе (например, /dev/ttyS0) или **auto** — для автоопределения. По умолчанию: auto.. Для драйвера хс параметр не используется.
- N633.** Скорость обмена данных для пульта №4. Для драйвера хс параметр не используется.
- N634.** Начальный адрес байта входов пульта №4.
- N635.** Начальный адрес байта выходов пульта №4.
- N636.** Тип управления пультом №4: «0» — пульт управляется драйвером; «1» — пульт управляется PLC.
- N637.** Имя файла конфигурации для пульта №4.
- N638.** Идентификатор модуля, к которому подключается драйвер пульта №4. Для драйвера snc11 — модуль sys, для драйвера хс — модуль usb.
- N639.** Тип физического подключения пульта №4. Для подключения через COM порт — com, для подключения через USB порт — usb. Для драйвера хс параметр не используется.
- N640.** Пульт №5 включен: «0» — выключен; «1» — включен.
- N641.** Имя драйвера, обрабатывающего данные пульта №5.
- N642.** Имя пульта №5, которым он представлен в файловой системе (например, /dev/ttyS0) или **auto** — для автоопределения. По умолчанию: auto.. Для драйвера хс параметр не используется.
- N643.** Скорость обмена данных для пульта №5. Для драйвера хс параметр не используется.
- N644.** Начальный адрес байта входов пульта №5.
- N645.** Начальный адрес байта выходов пульта №5.
- N646.** Тип управления пультом №5: «0» — пульт управляется драйвером; «1» — пульт управляется PLC.
- N647.** Имя файла конфигурации для пульта №5.
- N648.** Идентификатор модуля, к которому подключается драйвер пульта №5. Для драйвера snc11 — модуль sys, для драйвера хс — модуль usb.



- N649.** Тип физического подключения пульта №5. Для подключения через COM порт — com, для подключения через USB порт — usb. Для драйвера xhc параметр не используется.
- N700.** Количество каналов управляющих программ.
- N701.** Размер буфера управляющей программы канала №1.
- N711.** Размер буфера управляющей программы канала №2.
- N721.** Размер буфера управляющей программы канала №3.
- N902.** Номер вида отображения системы координат, используемый по умолчанию.
- N903.** Количество видов отображения системы координат.
- N910.** Угол поворота графики вокруг декартовой оси X для вида отображения №1.
- N911.** Угол поворота графики вокруг декартовой оси Y для вида отображения №1.
- N912.** Угол поворота графики вокруг декартовой оси Z для вида отображения №1.
- N920-N922.** Углы поворота графики вокруг декартовых осей для вида №2.
- N930-N932.** Углы поворота графики вокруг декартовых осей для вида №3.
- N940-N942.** Углы поворота графики вокруг декартовых осей для вида №4.
- N950-N952.** Углы поворота графики вокруг декартовых осей для вида №5.

2.2 Общие базовые станочные параметры

- N1000.** Количество осей координат. Для аналоговой версии: 1...6. Для цифровой версии: 1...8.
- N1025.** Ограничение времени, за которое ось должна переместиться в конечную точку кадра по обратной связи, если расчётная координата уже достигнута. Единица измерения: мс. По умолчанию: 1000.
- N1041.** Допускаемая погрешность разницы между начальным и конечным радиусами дуги в круговой интерполяции. Единица измерения: мм. По умолчанию: 0.01.



- N1042.** Допускаемая погрешность при определении координат центра дуги. Единица измерения: мкм. По умолчанию: 1000.
- N1501.** Отображать колонку номера магазина инструментов. По умолчанию: «0».
- N1502.** Отображать колонку номера слота в магазине инструментов. По умолчанию: «0».
- N1503.** Отображать колонку номера инструмента. По умолчанию: «1».
- N1504.** Отображать колонку номера привязки инструмента. По умолчанию: «1».
- N1505.** Отображать колонку названия инструмента. По умолчанию: «1».
- N1506.** Отображать колонку кода типа инструмента. По умолчанию: «0».
- N1507.** Отображать колонку значения угла инструмента. По умолчанию: «1».
- N1508.** Отображать колонку значения радиуса инструмента. По умолчанию: «1».
- N1509.** Отображать колонку значения радиуса кривизны инструмента. По умолчанию: «0».
- N1510.** Отображать колонку значения высоты инструмента. По умолчанию: «1».
- N1511.** Отображать колонку значения ширины инструмента. По умолчанию: «1».
- N1512.** Отображать колонку значения M-команды, исполняемой перед сменной инструмента. По умолчанию: «1».
- N1513.** Отображать колонку значения M-команды, исполняемой после смены инструмента. По умолчанию: «1».

2.3 Общие станочные параметры

- N2041.** Радиусная ось канала №1. В качестве значения можно указывать как название оси, так и её номер.
- N2051.** Радиусная ось канала №2. В качестве значения можно указывать как название оси, так и её номер.



N2061. Радиусная ось канала №3. В качестве значения можно указывать как название оси, так и её номер.

N2071. Радиусная ось канала №4. В качестве значения можно указывать как название оси, так и её номер.

2.4 Общие технологические параметры

N3000. Подача в конце кадра при G63, G64 (мм/мин), 0.1...1000 (5.0)

N3001. G-функция, действующая по умолчанию в группе G0-G3: 0...3 (1).

N3002. G-функция, действующая по умолчанию в группе G15-G16: 15...16.

N3003. G-функция, действующая по умолчанию в группе G17-G19: 17...19 (17).

N3004. G-функция, действующая по умолчанию в группе G221-G222: 221...222 (221).

N3005. G-функция, действующая по умолчанию в группе G223-G224: 223...224 (223).

N3006. G-функция, действующая по умолчанию в группе G40-G42: 40, 41, 42.

N3007. G-функция, действующая по умолчанию в группе G43-G44: 43, 44, 49.

N3008. G-функция, действующая по умолчанию в группе G53-G59: 53...59 (54).

N3009. G-функция, действующая по умолчанию в группе G61-G64 (62).

N3010. G-функция, действующая по умолчанию в группе G66-G67 (67).

N3011. G-функция, действующая по умолчанию в группе G90-G91: 90...91 (90).

N3012. G-функция, действующая по умолчанию в группе G94-G95: 94...95 (94).

N3013. G-функция, действующая по умолчанию в группе G96-G97: 96...97 (97).

N3020. Максимальная разница радиусов в начале и в конце дуги (мм) при круговой интерполяции 0...1.



- N3021.** Способ задания центра дуги окружности (I, J, K): «0» — в зависимости от G90.1/G91.1; «1» — в зависимости от G90/G91; «90» — всегда в абсолютных значениях; «91» — всегда в относительных значениях. По умолчанию: «0».
- N3022.** Тип интерполяции по G0: «0» — асинхронная интерполяция на максимальной скорости по каждой оси; «1» — линейная интерполяция на максимально возможной скорости.
- N3023.** Отключить стыковку скоростей между кадрами G0, если G0 используется в режиме интерполируемого перемещения: «0» — стыковать скорости; «1» — отключить стыковку. Значение по умолчанию: «1».
- N3030.** Диаметральное задание на I J K для токарных станков: «0» — не действует; «1» — действует.
- N3031.** Тип функции G92: «0» — задаются координаты в новой системе координат.
- N3033.** Единицы измерений аргументов тригонометрических функций: «0» — радианы; «1» — градусы.
- N3034.** Точность обхода углов при действии функции G50 (мм), 0...10.
- N3035.** Способ вывода коррекции на радиус инструмента, 0...2 (см. Программирование управляющих программ).
- N3042.** Тип компенсации на радиус инструмента: 0 — с автоматическим добавлением скруглений; 1 — по возможности без скруглений.
- N3050.** Режим обратной подачи при запрете процентовки подачи или при нарезании резьбы: 0 — обычная обратная подача с разгонами и торможениями, но пониженной точностью; 1 — обратная подача с разгонами и торможениями и дотягиванием в рамках кадра до позиции по датчику шпинделя. По умолчанию: 0.
- N3100.** Список M-команд, выполняемых до начала движения в кадре, очередность важна. M-функции разделяются двоеточием.
- N3101.** Список M-команд, выполняемых после окончания движения в кадре, очередность важна. M-команды разделяются двоеточием.
- N3102.** Список M-команд, выполняемых параллельно с движением в кадре, очередность важна. M-команды разделяются двоеточием.



- N3103.** Список M-команд, разрешённых на исполнение внутри коррекции на радиус. M-команды разделяются двоеточием. В данных M-командах запрещается модификация смещений системы координат. После исполнения любой из этих M-команд инструмент должен оказаться в той же позиции, что и на начало исполнения M-команды. Таким образом они либо не должны осуществлять перемещений, либо по окончании своей работы должны вернуть все оси в изначальные позиции.
- N3109.** Момент сброса модальных G-флагов: «00» — не сбрасывать, «01» — сбрасывать после завершения работы управляющей программы, «10» — сбрасывать перед стартом управляющей программы, «11» — сбрасывать перед стартом управляющей программы и сразу после её завершения.
- N3110.** Сброс G-кода группы №0 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3111.** Сброс G-кода группы №1 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3112.** Сброс G-кода группы №2 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3113.** Сброс G-кода группы №3 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3114.** Сброс G-кода группы №4 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3115.** Сброс G-кода группы №5 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3116.** Сброс G-кода группы №6 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3117.** Сброс G-кода группы №7 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3118.** Сброс G-кода группы №8 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3119.** Сброс G-кода группы №9 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3120.** Сброс G-кода группы №10 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.



- N3121.** Сброс G-кода группы №11 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3122.** Сброс G-кода группы №12 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3123.** Сброс G-кода группы №13 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3124.** Сброс G-кода группы №14 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3125.** Сброс G-кода группы №15 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3126.** Сброс G-кода группы №16 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3127.** Сброс G-кода группы №17 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3128.** Сброс G-кода группы №18 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3129.** Сброс G-кода группы №19 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3130.** Сброс G-кода группы №20 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3131.** Сброс G-кода группы №21 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3132.** Сброс G-кода группы №22 в начальное состояние: «0» — не сбрасывать; «2» — сбрасывать в значение по умолчанию.
- N3200.** Gxx — вызов G-функции как подпрограммы (№G функции): (номер программы). Например, 100:1000.1 — G100 вызовет подпрограмму 1000.нс.
- N3201-N3220.** Gxx — Вызов G-функции как подпрограммы например 100:1000.1 — G100 вызовет подпрограмму 1000.нс.
- N3500.** Имя оси безопасного подвода очередности 1 в канале 1. Имя должно совпадать с параметром N5n00.
- N3501.** Позиция оси безопасного подвода очередности 1 в канале 1.



- N3505.** Имя оси безопасного подвода очередности 2 в канале 1. Имя должно совпадать с параметром N5n00.
- N3506.** Позиция оси безопасного подвода очередности 2 в канале 1.
- N3510.** Имя оси безопасного подвода очередности 3 в канале 1. Имя должно совпадать с параметром N5n00.
- N3511.** Позиция оси безопасного подвода очередности 3 в канале 1.
- N3515.** Имя оси безопасного подвода очередности 4 в канале 1. Имя должно совпадать с параметром N5n00.
- N3516.** Позиция оси безопасного подвода очередности 4 в канале 1.
- N3550.** Имя оси безопасного подвода очередности 1 в канале 2. Имя должно совпадать с параметром N5n00.
- N3551.** Позиция оси безопасного подвода очередности 1 в канале 2.
- N3555.** Имя оси безопасного подвода очередности 2 в канале 2. Имя должно совпадать с параметром N5n00.
- N3556.** Позиция оси безопасного подвода очередности 2 в канале 2.
- N3560.** Имя оси безопасного подвода очередности 3 в канале 2. Имя должно совпадать с параметром N5n00.
- N3561.** Позиция оси безопасного подвода очередности 3 в канале 2.
- N3565.** Имя оси безопасного подвода очередности 4 в канале 2. Имя должно совпадать с параметром N5n00.
- N3566.** Позиция оси безопасного подвода очередности 4 в канале 2.
- N3650.** Указанные через запятую номера групп M-команд (нумерация с 1), команды из которых должны исполняться вместе с T-командой, то есть привязанные к инструменту. Используются вместе с продолжением строки, подробнее см. N3660.
- N3651.** Указанные через запятую номера групп M-команд (нумерация с 1), команды из которых должны исполняться вместе с S-командой, то есть привязанные к шпинделю. Используются вместе с продолжением строки, подробнее см. N3660.



N3660...N3699. Группа М-команд. Группирует взаимоисключающие М-команды, одновременно из числа которых может работать только одна. Команды из таких групп автоматически исполняются перед продолжением со строки (функция возврата на контур), если были найдены ранее в программе. Исполняется последняя найденная из каждой группы.

2.5 Общие наладочные параметры

N4001. Минимальное значение корректора F, в процентах: 0...100. По умолчанию: 0%.

N4002. Максимальное значение корректора F, в процентах: 100...255. По умолчанию: 120%.

N4003. Минимальное значение корректора S, в процентах: 0...100. По умолчанию: 40%.

N4004. Максимальное значение корректора S, в процентах: 100...255. По умолчанию: 140%.

N4005. Разрешить коррекцию подачи быстрого хода (G0). Используется в случаях, когда необходимо корректировать подачу рабочего хода, не изменяя подачу быстрого хода: «0» — коррекция подачи быстрого хода осуществляется корректором подачи рабочего хода; «1» — разрешить коррекцию. По умолчанию: «0».

N4006. Минимальное значение корректора подачи быстрого хода, в процентах: 0...100. По умолчанию: 0%.

N4007. Максимальное значение корректора подачи быстрого хода, в процентах: 100...255. По умолчанию: 100%.

2.6 Базовые станочные параметры оси

Параметры N5000...N5046 описывают базовые станочные параметры оси №1. Описание базовых станочных параметров осей 2-8 повторяет описание соответствующих параметров оси №1.

N5000. Имя оси, до двух латинских символов (например: X, Y, Z, XA, XB, U, W и т. п.).

N5002. Тип датчика обратной связи по положению: «0» — инкрементальный; «1» — абсолютный. По умолчанию: «0».



- N5003.** Разрешить проверку программных конечных выключателей: «0» — запретить; «1» — разрешить.
- N5004.** Тип выхода в «0» по оси: «1» — только по 0-метке; «2» — поиск нулевого концевика, затем по 0-метке; «3» — только по нулевому концевику. По умолчанию: «2».
- N5005.** Номер канала оцифровки координаты оси, 0...8: «0» — у оси нет оцифровки. Каналом считается сквозной порядковый номер датчика обратной связи по всем драйверам, указанных в параметрах.
- N5006.** Номер канала ЦАП координаты оси, 0...8: «0» — у оси нет ЦАП. Каналом считается сквозной порядковый номер ЦАП/привода по всем драйверам, указанных в параметрах.
- N5007.** Направление начала выхода в «ноль» по оси: «-1» — отрицательное направление; «1» — положительное направление; «0» — направление задаётся модулем PLC.
- N5008.** Требуется ли выход в «ноль» по оси: «-1» — выход в «ноль» запрещён; «0» — выход в «ноль» не требуется; «1» — выход в «ноль» обязателен.
- N5009.** Реверс выхода в «0»: «0» — при съезде с нулевого концевика руководствоваться параметром N5n37; «1» — при наезде на нулевой концевик съезд происходит в обратную сторону (позиция «нуля» может отличаться при наезде на концевик с разных сторон). Параметр устарел и будет удалён в одной из следующих версий ЧПУ.
- N5010.** Программное ограничение перемещения в «+» по оси (мм, град), -100000...100000.
- N5011.** Программное ограничение перемещения в «-» по оси (мм, град), -100000...100000.
- N5012.** Подача быстрого хода оси (мм/мин), 1...100000. На этой подаче происходит позиционирование оси по функции G0.
- N5013.** Знаменатель коэффициента приведения показаний датчика обратной связи к реальному, 0...65535.
- N5014.** Числитель коэффициента приведения показаний датчика обратной связи к реальному, -32767...32767.
- N5015.** Смещение нуля станка, на которое необходимо переместиться по окончанию выхода в «0» (мм).



- N5016.** Позиция нуля станка, которая будет присвоена нулю станка по окончанию выхода в «0» (мм).
- N5021.** Тип дискретности оси: «0» — радиусная; «1» — диаметральной дискретность текущего положения. Диаметральной дискретность означает, что при отработке УП заданная координата по оси будет делиться на 2.
- N5022.** Номер оси в декартовой системе координат для определения плоскости (G17, G18, G19): «1» — декартова ось X; «2» — декартова ось Y; «3» — декартова ось Z; «0» — нет соответствия декартовым осям.
- N5024.** Количество импульсов на один оборот датчика обратной связи (необходимо для определения скорости вращения). Для датчиков, подключенных напрямую к плате ЧПУ, данное значение будет соответствовать дискретности датчика, домноженной на 4. Для цифровых датчиков (EtherCAT) указывается оригинальное значение.
- N5030.** Номер канала, в котором используется ось: 1...3, «0» — ось может использоваться во всех каналах.
- N5031.** Тип оси: «0» — обычная ось; «1» — шпиндель.
- N5032.** «0» — линейная ось; «1» — поворотная ось (необходимо задать для использования интерполяции в полярных координатах).
- N5036.** Позиция для отображения оси в интерфейсе пользователя, 1-8 («0» — не отображать). Допускается указание одной и той же позиции для разных осей, чтобы поочередно менять их местами из модуля ПЛК (обменная ячейка AX1_INACTIVE_O).
- N5037.** Направление съезда с нулевого концевика: «-1» — отрицательное; «1» — положительное; «0» — в зависимости от направления движения. Работает совместно с параметром реверса.
- N5040.** Альтернативное имя оси, используемое для перемещения по ней в режиме G91.
- N5041.** Флаг виртуальной оси: «0» — обычная ось; «1» — виртуальная ось.
- N5042.** Флаг подчинённой оси: «0» — ось самостоятельная; «1» — ось является подчинённой. По умолчанию: «0». Параметр работает совместно с параметром N5043.
- N5043.** Номер главной оси, с которой будет синхронизироваться движение данной оси. Параметр работает совместно с параметром N5042.



N5044. Максимальное рассогласование между данной осью и главной осью, с которой производится синхронизация (если указан параметр N5042). Единица измерения: мм. Если указать значение «0», контроль рассогласования между синхронизируемыми осями будет отключен. По умолчанию: 10.0.

N5046. Режим теневого смещения позиции оси из ПЛК: «0» — интерполируемое смещение оси; «1» — грубое смещение координаты без учёта разгона и торможения по оси. В режиме теневого смещения ПЛК может изменять физическое положение оси без изменения расчётных координат.

N5050. Флаг служебной оси. Служебная ось не может управляться по программе, она предназначена для управления из ПЛК. 1 — служебная ось; 0 — ось может управляться по программе. По умолчанию: 0.

N5100-N5146. Базовые станочные параметры оси №2.

N5200-N5246. Базовые станочные параметры оси №3.

N5300-N5346. Базовые станочные параметры оси №4.

N5400-N5446. Базовые станочные параметры оси №5.

N5500-N5546. Базовые станочные параметры оси №6.

N5600-N5646. Базовые станочные параметры оси №7.

N5700-N5746. Базовые станочные параметры оси №8.

N5800-N5846. Базовые станочные параметры оси №9.

N5900-N5946. Базовые станочные параметры оси №10.

N10000-N10046. Базовые станочные параметры оси №11.

N10100-N10146. Базовые станочные параметры оси №12.

2.7 Станочные параметры оси

Параметры N6000...N6046 описывают станочные параметры оси №1. Описание станочных параметров осей 2-8 повторяет описание соответствующих параметров оси 1.

N6000. Ускорение рабочих перемещений (мм/мин/сек), 1...100000. Параметр определяет интенсивность увеличения величины подачи оси.



- N6001.** Ускорение аварийных перемещений (мм/мин/сек), 1...100000. Параметр определяет интенсивность изменения подачи оси в экстренных ситуациях (например, наезд на концевой ограничитель). «0» — взять значение ускорения нормального перемещения (см. N6000).
- N6002.** Ускорение быстрых перемещений (мм/мин/сек), 1...100000. Параметр определяет интенсивность изменения подачи оси на быстрых перемещениях (например, по G0). «0» — взять значение ускорения (см. N6000).
- N6003.** Замедление рабочих перемещений (мм/мин/сек), 1...100000. Параметр определяет интенсивность уменьшения величины подачи оси. «0» — взять значение ускорения (см. N6000)
- N6004.** Замедление аварийных перемещений (мм/мин/сек), 1...100000. «0» — взять значение ускорения при аварийных перемещениях (см. N6001).
- N6005.** Замедление быстрых перемещений (мм/мин/сек), 1...100000. Параметр определяет интенсивность изменения подачи оси на быстрых перемещениях (например, по G0). «0» — взять значение ускорения по G0 (см. N6002).
- N6006.** Подача в режиме выхода в ноль (мм/мин), 1...50000.
- N6007.** Максимальное рассогласование (мм), 0.001...60. При превышении установленного значения выдается сообщение об ошибке (сбой рассогласования оси).
- N6008.** Добротность, -200...200. Знак определяет знак задания на привод, нормально 0.1.
- N6009.** Коэффициент скоростной компенсации, 0.0...1.0. Если скоростной коэффициент равен 1.0, то в качестве управляющего задания по скорости будет подаваться значение скорости целиком. Если скоростной коэффициент равен 0, то управление по скорости не осуществляется.
- N6010.** Зона контроля (мм), 0.001...1.0. Определяет рассогласование, при достижении которого в режиме точного перемещения можно приступить к исполнению следующего кадра.
- N6011.** Разрешить компенсацию дрейфа привода: «0» — запретить; «1» — разрешить. По умолчанию: «1». Дрейф компенсируется в состоянии покоя оси, как только значение датчика обратной связи стабилизируется.



- N6012.** Интегрировать остаток задания на ЦАП: «0» — остаток задания отбрасывается (для приводов с обратной связью); «1» — остаток задания прибавляется к заданию в следующем такте (для приводов без обратной связи). По умолчанию: «0».
- N6013.** Подача в режиме выхода в «ноль» на конечных выключателях точного ноля и в режиме поиска 0-метки (см. AX1_LIM_HOME_O) (мм/мин), 1...10000.
- N6015.** Максимально допустимая скорость перемещения по оси, мм/мин. Значение необходимо выставлять большим либо равным значению параметра N5012. Если процентовой скорости допускается превышение 100% быстрой подачи, то в данном параметре необходимо указать скорость, соответствующую максимальному значению процентки скорости. Значение «0» соответствует отсутствию ограничения. По умолчанию: «0».
- N6016.** Интервал времени в секундах, в течение которого рассогласование может превышать заданное, либо в течение которого скорость может превышать максимальную, 0.0...10.0.
- N6017.** Зона нечувствительности (мёртвая зона) для управления приводом (мкм), 0...32000. Данный параметр можно использовать для компенсации постоянного качения оси (например, при наличии маленького люфта).
- N6019.** Коэффициент корректировки ускорения, 0.0...1.0. Определяет добавочное ускорение, формируемое из заданного ускорения.
- N6020.** Ограничение рассогласования, через которое осуществляется управление скоростью по положению. По умолчанию: «0».
- N6021.** Пропорциональный коэффициент ПИД-регулятора, 0.0...1.0. По умолчанию: 0.17.
- N6022.** Интегральный коэффициент ПИД-регулятора, 0.0...0.1. По умолчанию: 0.0.
- N6023.** Дифференциальный коэффициент ПИД-регулятора, 0.0...10.0. По умолчанию: 0.0.
- N6025.** Коэффициент коррекции изменения ускорения, 0.0...10.0.
- N6026.** Смещение задания на ЦАП, $-10.0 \dots + 10.0$.



- N6028.** Коэффициент П-регулятора управления по положению наперёд, 0.0...1.0. Можно использовать вместо ПИД-регулятора для управления осью по рассогласованию.
- N6029.** Значение компенсации неинтерполируемого люфта.
- N6032.** Ограничение максимального количество импульсов, выдаваемых датчиком обратной связи оси по положению за один такт реального времени (1 мс), 0...1000000. При превышении выдаётся ошибка (сбой превышения ограничения импульсов по датчику обратной связи по положению).
- N6033.** Включить контроль превышения задания на ЦАП: «0» — выключен; «1» — разрешён.
- N6034.** Рассогласование интерполируемого люфта в мм, при достижении которого люфт считается пройденным, 0.0...10.0.
- N6035.** Скорость прохождения интерполируемого люфта, мм/мин.
- N6036.** Коэффициент ПИД-регулятора, добавляющий пропорциональную составляющую квадратного корня от рассогласования, 0.0...1.0.
- N6037.** Коэффициент ПИД-регулятора, добавляющий интегральную составляющую квадратного корня от рассогласования, 0.0...0.1.
- N6038.** Коэффициент ПИД-регулятора, добавляющий дифференциальную составляющую квадратного корня от рассогласования, 0.0...1.0.
- N6039.** Коэффициент ПИД-регулятора, добавляющий пропорциональную составляющую квадрата от рассогласования, 0.0...1.0.
- N6040.** Коэффициент ПИД-регулятора, добавляющий интегральную составляющую квадрата от рассогласования, 0.0...0.1.
- N6041.** Коэффициент ПИД-регулятора, добавляющий дифференциальную составляющую квадрата от рассогласования, 0.0...1.0.
- N6042.** Наброс на ЦАП при движении в «-». По умолчанию: 0.0.
- N6043.** Наброс на ЦАП при движении в «+». По умолчанию: 0.0.
- N6044.** Минимальное задание на ЦАП по модулю, при котором осуществляется наброс, указанный в параметрах **N6042** и **N6043**. По умолчанию: 0.0.



N6045. Флаг включения ПИД-регулятора при обработке интерполируемого люфта: «0» — оставить ПИД-регулятор включённым; «1» — выключать ПИД-регулятор. По умолчанию: «0».

N6063. Ускорение перемещений по маховику (мм/мин/сек), 1...300000. «0» — взять значение ускорения нормального перемещения (см. N6000).

N6064. Замедление перемещений по маховику (мм/мин/сек), 1...300000. «0» — взять значение ускорения при перемещениях в режиме маховика (см. N6063).

N6064. Мягкое ускорение, способ ограничения итоговой скорости, выдаваемой в привод, по отношению к предыдущей скорости. Может использоваться для смягчения теневых перемещений совместно с основными. 0 — разница между новой скоростью и предыдущей может превышать ускорение аварийного движения, 1 — новая итоговая скорость ограничивается аварийным ускорением по отношению к предыдущей итоговой скорости.

N6100-N6164. Станочные параметры оси 2.

N6200-N6264. Станочные параметры оси 3.

N6300-N6364. Станочные параметры оси 4.

N6400-N6464. Станочные параметры оси 5.

N6500-N6564. Станочные параметры оси 6.

N6600-N6664. Станочные параметры оси 7.

N6700-N6764. Станочные параметры оси 8.

N6800-N6864. Станочные параметры оси 9.

N6900-N6964. Станочные параметры оси 10.

N11000-N11064. Станочные параметры оси 11.

N11100-N11164. Станочные параметры оси 12.

2.8 Наладочные параметры оси

Параметры N7000...N7009 описывают наладочные параметры оси №1. Описание наладочных параметров осей 2–8 повторяет описание соответствующих параметров оси №1.



- N7000.** Смещение нуля координаты относительно нуля по конечным выключателям для 1-й оси, мм/град (-999999...+999999).
- N7002.** 2-я фиксированная точка координаты, в которую происходит переход по G30 по 1-й оси, мм (-999999...+999999).
- N7003.** 3-я фиксированная точка координаты, в которую происходит переход по G30 по 1-й оси, мм (-999999...+999999).
- N7004.** 4-я фиксированная точка координаты, в которую происходит переход по G30 по 1-й оси, мм (-999999...+999999).
- N7005.** 5-я фиксированная точка координаты, в которую происходит переход по G30 по 1-й оси, мм (-999999...+999999).
- N7006.** 6-я фиксированная точка координаты, в которую происходит переход по G30 по 1-й оси, мм (-999999...+999999).
- N7007.** 7-я фиксированная точка координаты, в которую происходит переход по G30 по 1-й оси, мм (-999999...+999999).
- N7008.** 8-я фиксированная точка координаты, в которую происходит переход по G30 по 1-й оси, мм (-999999...+999999).
- N7009.** 9-я фиксированная точка координаты, в которую происходит переход по G30 по 1-й оси, мм (-999999...+999999).
- N7100-N7109.** Наладочные параметры оси №2.
- N7200-N7209.** Наладочные параметры оси №3.
- N7300-N7309.** Наладочные параметры оси №4.
- N7400-N7409.** Наладочные параметры оси №5.
- N7500-N7509.** Наладочные параметры оси №6.
- N7600-N7609.** Наладочные параметры оси №7.
- N7700-N7709.** Наладочные параметры оси №8.
- N7800-N7809.** Наладочные параметры оси №9.
- N7900-N7909.** Наладочные параметры оси №10.
- N12000-N12009.** Наладочные параметры оси №11.
- N12100-N12109.** Наладочные параметры оси №12.



2.9 Параметры пользователя

N8001–N8050. Возможен ввод дробных значений. Доступ к параметрам возможен из языка макропрограммирования и из программы электроавтоматики PU. Назначение параметров зависит от конкретного станка и обычно отражается в документации на станок.

N8501–N8550. Возможен ввод только целых значений. Доступ к параметрам возможен из языка макропрограммирования и из программы электроавтоматики через системные параметры пользователя P. Назначение параметров зависит от конкретного станка и обычно отражается в документации на станок.

2.10 Базовые станочные параметры маховика

N9001. Знаменатель коэффициента приведения показаний маховиков по умолчанию, 0...65535.

N9002. Числитель коэффициента приведения показаний маховиков по умолчанию, -32767...32767.

Параметры N9011...N9016 описывают параметры маховика №1. Параметры маховиков 2–9 повторяют соответствующие параметры маховика №1.

N9011. Знаменатель коэффициента приведения показаний маховика №1, 0...65535.

N9012. Числитель коэффициента приведения показаний маховика №1, -32767...32767.

N9013. Роль маховика: «0» — отключен; «1» — произвольное правление; «2» — управление осью; «3» — управление процентовкой скорости; «4» — управление процентовкой шпинделя; «5» — автоматическое управление драйвером устройства.

N9014. Управление стандартной осью: «0» — отключен; «1» — стандартной осью X; «2» — стандартной осью Y; «3» — стандартной осью Z; «4» — стандартной осью A; «5» — стандартной осью B; «6» — стандартной осью C (шпиндель в режиме оси).

N9015. Номер управляемой оси. Если указать «0», управление будет автоматическое исходя из параметров N9013 и N9014.



N9016. Номер входного канала энкодера (оцифровки). Если указать «0», маховик будет отключен.

N9021-N9026. Базовые станочные параметры маховика №2.

N9031-N9036. Базовые станочные параметры маховика №3.

N9041-N9046. Базовые станочные параметры маховика №4.

N9051-N9056. Базовые станочные параметры маховика №5.

N9061-N9066. Базовые станочные параметры маховика №6.

N9071-N9076. Базовые станочные параметры маховика №7.

N9081-N9086. Базовые станочные параметры маховика №8.

N9091-N9096. Базовые станочные параметры маховика №9.

2.11 Настройка модулей

Для настройки модуля управления используются параметры N410–N418:

- **N410, N413, N416** — имя модуля (по умолчанию **sys**);
- **N411, N414, N417** — имя файла конфигурации (в настоящий момент ни один из модулей не использует данный параметр);
- **N412, N415, N418** — номер основного устройства (в настоящий момент ни один из модулей не использует данный параметр).

В системе доступны следующие модули.

- Модуль **ethercat** используется совместно с устройствами, общающимися с ЧПУ через протокол **EtherCAT**.
- Модуль **sys** используется совместно с устройствами, общающимися с ЧПУ через шину **ISA**.
- Модуль **usb** используется совместно с нестандартными устройствами, общающимися с ЧПУ через порт **USB**.
- Модуль **titan** используется совместно с устройствами, общающимися с ЧПУ через плату **link11**.



В дальнейшем в параметрах указывается идентификатор модуля, а не его имя. Идентификатор основан на порядке следования параметров. Модуль, прописанный в параметре N410, будет иметь идентификатор 0; модуль, прописанный в параметре N413, будет иметь идентификатор 1; модуль, прописанный в параметре N416, будет иметь идентификатор 2. Для подключения какого-либо модуля достаточно прописать его имя в один из параметров: N410, N413 или N416.

2.11.1 Настройка драйвера блока ЦАП

Для блоков ЦАП существуют три драйвера: **ethercat**, **cnc11** и **titan**. Каждый драйвер подключается к своему модулю. Алгоритм настройки драйвера ЦАП с порядковым номером n (1–4).

1. Подключаем в ЧПУ нужный модуль (для драйвера **ethercat** — модуль **ethercat**; для драйвера **cnc11** — модуль **sys**; для драйвера **titan** — модуль **titan**), прописав его в одном из параметров: N410, N413 или N416.
2. Прописываем идентификатор нужного модуля в параметр $Nn1$.
3. Прописываем имя нужного драйвера в параметр $Nn2$.
4. Настраиваем параметр $Nn0$ в соответствии с требованиями драйвера.
5. Выставляем необходимое количество каналов в параметр $Nn3$.

2.11.2 Настройка драйвера блока оцифровки

Для блоков оцифровки существуют три драйвера: **ethercat**, **cnc11**, **titan**. Каждый драйвер подключается к своему модулю. Алгоритм настройки драйвера оцифровки с порядковым номером n (5–8).

1. Подключаем в ЧПУ нужный модуль (для драйвера **ethercat** — модуль **ethercat**; для драйвера **cnc11** — модуль **sys**; для драйвера **titan** — модуль **titan**), прописав его в одном из параметров: N410, N413 или N416.
2. Прописываем идентификатор нужного модуля в параметр $Nn1$.
3. Прописываем имя нужного драйвера в параметр $Nn2$.
4. Настраиваем параметр $Nn0$ в соответствии с требованиями драйвера.
5. Выставляем необходимое количество каналов в параметр $Nn3$.
6. Выставляем разрядность блока оцифровки параметр $Nn4$ (по умолчанию для всех драйверов используется разрядность 24).



2.11.3 Настройка входов

Для блоков входов существуют три драйвера **ethercat**, **sys**, **titan**. Каждый драйвер подключается к своему модулю. Алгоритм настройки входов с порядковым номером n (1-10).

1. Подключаем в ЧПУ нужный модуль (для драйвера **ethercat** — модуль **ethercat**; для драйвера **snc11** — модуль **sys**; для драйвера **titan** — модуль **titan**), прописав его в одном из параметров: N410, N413 или N416.
2. Подключаем необходимый драйвер, прописав его имя в одном из параметров: N421, N423 или N425, а идентификатор нужного модуля прописать соответственно в одном из параметров N420, N422 или N424.
3. Прописываем идентификатор нужного драйвера в параметр $N[203+5*(n-1)]$.
4. Настраиваем параметры $N[200+5*(n-1)]$, $N[201+5*(n-1)]$, $N[202+5*(n-1)]$, $N[204+5*(n-1)]$ согласно требованиям драйвера и характеристикам плат входов.
5. Для драйвера **ethercat** необходимо записать имя конфигурационного файла платы входов/выходов в параметр $N[430+(n-1)]$.
6. Для драйвера **titan** необходимо записать имя устройства, через которое происходит считывание плат входов/выходов в параметр $N[440+(n-1)]$.

Параметры N430–N449 применяются для группы входов/выходов.

2.11.4 Настройка выходов

Для блоков выходов существуют три драйвера **ethercat**, **sys**, **titan**. Каждый драйвер подключается к своему модулю. Алгоритм настройки выходов с порядковым номером n (1-10).

1. Подключаем в ЧПУ нужный модуль (для драйвера **ethercat** — модуль **ethercat**; для драйвера **snc11** — модуль **sys**; для драйвера **titan** — модуль **titan**), прописав его в одном из параметров: N410, N413 или N416.
2. Подключаем необходимый драйвер, прописав его имя в одном из параметров: N421, N423 или N425, а идентификатор нужного модуля прописать соответственно в одном из параметров N420, N422 или N424.
3. Прописываем идентификатор нужного драйвера в параметр $N[303+5*(n-1)]$.



4. Настраиваем параметры $N[300+5*(n-1)]$, $N[301+5*(n-1)]$, $N[302+5*(n-1)]$, $N[304+5*(n-1)]$ согласно требованиям драйвера и характеристикам плат выходов.
5. Для драйвера ethercat необходимо записать имя конфигурационного файла платы входов/выходов в параметр $N[430+(n-1)]$.
6. Для драйвера titan необходимо записать имя устройства, через которое происходит считывание плат входов/выходов в параметр $N[440+(n-1)]$.

Параметры N430–N449 применяются для группы входов/выходов.

2.12 Настройка параметров для пульта оператора

2.12.1 Настройка пульта оператора spc11

Пример настройки пульта оператора spc11 (СПО24/3) подключенного через СОМ-порт.

1. Включаем обработку пульта, выставляя параметр **N6n0** в 1.
2. Подключаем драйвер обработки данных пульта **N6n1**. Для пульта spc11 (СПО24/3) используется драйвер spc11.
3. В параметр **N6n2** прописываем «/dev/ttyS0». Устройства, подключенные в СОМ-порт, отображаются в файловой системе как «/dev/ttyS0». Имя определяется очередностью подключения устройств. Нумерация начинается с 0.
4. Выставляем параметр **N6n3** в 19200 (стандартная скорость обмена).
5. В параметр **N6n4** прописываем 201, что является стандартным адресом байтов входов для пульта №1.
6. В параметр **N6n5** прописываем 201, что является стандартным адресом байтов выходов для пульта №1.
7. В параметр **N6n6** прописываем 0, т. к. драйвер spc11 не поддерживает управление от PLC.
8. Параметр **N6n7** в драйвере spc11 не используется.
9. В параметр **N6n8** прописываем идентификатор модуля, к которому подключается драйвер пульта №(n+1). Например, если параметр **N410** равен sys, то идентификатор модуля будет равен 0.



10. В параметр **N6n9** прописываем тип физического подключения пульта №1 в данном примере это com.

2.12.2 Настройка пульта оператора НВ-04

Пример настройки пульта оператора НВ-04.

1. Включаем обработку пульта, выставляя параметр **N6n0** в 1.
2. Подключаем драйвер обработки данных пульта **N6n1**. Для пульта НВ-04 используется драйвер xhc.
3. Параметр **N6n2** в драйвере xhc не используется.
4. Параметр **N6n3** в драйвере xhc не используется.
5. В параметр **N6n4** прописываем 301, что является стандартным адресом байтов входов для пульта №2.
6. В параметр **N6n5** прописываем 301, что является стандартным адресом байтов выходов для пульта №2.
7. В параметр **N6n6** прописываем 0, т. к. драйвер xhc не поддерживает управление от PLC.
8. В параметр **N6n7** прописываем имя файла конфигурации. Стандартное имя: «console_xhc.json».
9. В параметр **N6n8** прописываем идентификатор модуля, к которому подключается драйвер пульта №(n+1). Например, если параметр **N413** равен usb, то идентификатор модуля будет равен 1.
10. Параметр **N6n9** в драйвере xhc не используется.

Пример файла конфигурации для пульта оператора НВ-04.

```
1 {
2   "encoders": {"encoder": [{"role": "AUTO", "axis": "X", "nominator": 1, "denominator": 1}]},
3   "indication": {
4     "channels": [
5       {
6         "axis": [
7           {"role": "X", "cs": "M", "channel": 0},
8           {"role": "X", "cs": "W", "channel": 0},
```



```
9         {"role": "Y", "cs": "M", "channel": 0},
10        {"role": "Y", "cs": "W", "channel": 0},
11        {"role": "Z", "cs": "M", "channel": 0},
12        {"role": "Z", "cs": "W", "channel": 0},
13        {"role": "A", "cs": "M", "channel": 0},
14        {"role": "A", "cs": "W", "channel": 0}
15    ]
16 }
17 ]
18 }
19 }
```

1. **encoders** — описание энкодеров пульта.
2. **encoders.encoder** — описание энкодера пульта.
3. **encoders.encoder.role** — описание роли энкодера, может принимать значения:
 - (a) **OFF** — энкодер выключен;
 - (b) **AXIS** — энкодер управляет только той осью, которая прописана в параметре axis;
 - (c) **F** — энкодер управляет только процентровкой подачи;
 - (d) **S** — энкодер управляет только процентровкой шпинделя;
 - (e) **AUTO** — роль энкодера зависит от переключателя на пульте.
4. **indication** — описание индикации осей на удаленном пульте.
5. **indication.channels** — описание нескольких суппортов. Если суппортов несколько, то описание индикации суппортов происходит в фигурных скобках, разделенных запятой.
6. **indication.channels.axis** — описание выводимых осей на индикацию.
7. **indication.channels.axes** — имя оси.
8. **indication.channels.role** — роль оси в декартовой системе координат.
9. **indication.channels.none** — ось не выводится на индикацию.
10. **indication.channels.cs** — система координат:
 - (a) **M** — машинная система координат;
 - (b) **W** — рабочая система координат.



11. **indication.channels.channel** — к какому каналу принадлежит ось.

Порядок индикации осей:

- слева сверху,
- справа сверху,
- снизу слева,
- снизу справа.

2.13 Перечень параметров, необходимых для организации управления приводами подачи

Для каждой оси с помощью делителя (параметр **N5n13**) и множителя (параметр **N5n14**), указанных в параметрах оси, формируется минимальная величина программируемого перемещения.

По умолчанию используется метрическая система координат. В параметрах необходимо задать соответствие между минимальной величиной перемещения и величиной перемещения в 1 мкм.

Для корректного перемещения оси в режиме управления по скорости необходимо также правильно выставить добротность привода (параметр **N6n08**). То есть расстояние, которое проедет привод без включённого ПИД-регулятора, должно наиболее точно соответствовать расстоянию, полученному с энкодера двигателя. Для цифровых приводов идеальное значение добротности соответствует числу ± 16.66666666666667 (± 16 . (6)).

Скорость форсированной продольной подачи (в команде G00) устанавливается изготовителем станка для каждой оси отдельно в параметре **N5n12**. В режиме позиционирования в начале блока происходит ускорение инструмента до предварительно заданной скорости, а в конце блока – замедление. Программа переходит к выполнению следующего блока после подтверждения выхода в заданную позицию. «Выход в заданную позицию» означает, что двигатель подачи находится в заданном диапазоне. Этот диапазон устанавливается изготовителем станка в параметре **N6n10**.

Подача — скорость движения инструмента. При движении подача раскладывается на составляющие по каждой из осей. Скорость движения инструмента может быть вычислена при помощи формулы 2.13.1.

$$F = \sum_{i=1}^N F_i = F_1 + F_2 + \dots + F_N \quad (2.13.1)$$



Существует два вида подачи: ускоренная подача и подача резания.

- Ускоренная подача предназначена для позиционирования инструмента на максимальной скорости в нужную точку.
- Подача резания используется при обработки деталей и обычно намного ниже ускоренной, чтобы обеспечить большую точность обработки детали. Данный вид подачи задаётся в управляющей программе.

Подача представляет из себя скорость движения инструмента. Но привода либо не могут мгновенно начать движение с заданной скоростью вследствие нелинейных характеристик двигателей, либо будут приводить к жёсткому механическому удару. Поэтому происходит автоматическое ускорение в начале движения и автоматическое замедление в конце движения. Время ускорения и время замедления представляют из себя довольно маленькие величины, которые определяются параметрами **N6n00** для подачи резания и **N6n01** для ускоренной подачи.

2.14 Настройка осей

2.14.1 Дискретность оси для круговых датчиков

Дискретность оси задаётся множителем и делителем, которые приводят показания с датчика к 1 мкм перемещения.

Расчёт дискретности датчика для случая, когда датчик установлен напрямую на ШВП оси, либо на шестерне реечной передачи:

$$\frac{N5n14}{N5n13} = \frac{P \cdot 1000}{N}$$

Расчёт дискретности датчика для случая, когда датчик установлен на валу двигателя, подключенного через редуктор:

$$\frac{N5n14}{N5n13} = \frac{P \cdot 1000}{N} \cdot \frac{1}{K_P}$$

Расчёт дискретности датчика для случая, когда датчик установлен отдельно от оси, через редуктор:

$$\frac{N5n14}{N5n13} = \frac{P \cdot 1000}{N} \cdot \frac{1}{K_N}$$

При использовании реечной передачи расстояние P можно рассчитать через модуль шестерни и количество зубьев:

$$P = m\pi z$$



Ниже приведена расшифровка обозначений, использованных в формулах.

- P — расстояние (в миллиметрах) или угол (в градусах), на которое перемещается ось за один оборот ШВП (шаг ШВП) либо за один оборот шестерни, если используется реечная передача.
- N — количество импульсов на один оборот датчика (при подключении к аналоговой версии ЧПУ вычисляется как произведение дискретности датчика на 4).
- K_P — коэффициент редуктора двигателя (для редуктора 10:1 $K_P = \frac{10}{1}$), соответствующий количеству оборотов двигателя на 1 оборот ШВП либо на 1 оборот шестерни в случае реечной передачи.
- K_N — коэффициент редуктора датчика (для редуктора 10:1 $K_N = \frac{10}{1}$), соответствующий количеству оборотов датчика на 1 оборот ШВП либо на 1 оборот шестерни в случае реечной передачи.
- m — модуль шестерни в случае реечной передачи.
- z — количество зубьев у шестерни в случае реечной передачи.

Полученная дробь дискретности оси должна быть записана в целых числах, а у числителя и знаменателя не должно быть общих делителей, то есть дробь должна быть сокращена по всем общим делителям. *В противном случае, при очень больших значениях множителя может происходить переполнение значения при умножении на показания датчиков с большой дискретностью.*

2.14.2 Настройка выхода в 0

Тип выхода в 0 задаётся параметром **N5n04**. В зависимости от типа меняется алгоритм поиска 0 станка.

Если ось поддерживает поиск 0-метки, то рекомендуется выбирать выход в 0 по конечному концевiku с дальнейшим поиском 0-метки. При этом необходимо убедиться, что позиция 0-метки не находится предельно близко к конечному концевiku, иначе по съезду с концевика может находиться как целевая, так и следующая 0-метка.

Выход в 0 по концевiku может понадобиться для осей без обратной связи, но в этом случае возникает погрешность, зависящая от времени такта сверхбыстрой секции ПЛК и от времени отклика плат входов/выходов, через которые считываются сигналы с датчиков.



Выход в 0 только по 0-метке можно использовать у поворотных осей с передаточным соотношением 1:1, либо у шпинделя, если он переключен в режим оси.

Если по оси выход в 0 обязателен, то необходимо выставить параметр **N5n08**, чтобы предотвратить исполнение УП без выхода в 0 по осям. Случайное исполнение УП без выхода в 0 может привести к аварии или поломке инструмента.

Направление выхода можно задавать как параметром **N5n07**, так и из ПЛК с помощью функции **HOMING_START**. При этом значение из параметра является приоритетным. Также необходимо выставить направление съезда с концевика с помощью параметра **N5n37**. Например, если съезд с концевика выбран в «+», то, независимо от того, с какой стороны ось наехала на концевик, съезд с него всегда будет в положительном направлении.

Поиск концевика идет на скорости, заданной параметром **N6n06**, а съезд с концевика и поиск 0-метки осуществляется на скорости, заданной параметром **N6n13**.

Если во время поиска нулевого концевика происходит наезд на концевой ограничитель движения, ось меняет направление движения и начинает поиск нулевого концевика в изменённом направлении.

Алгоритм выхода в «0» по нулевому концевика и 0-метке:

1. Ось движется с текущей позиции в направлении выхода в 0.
2. Если найден концевой ограничитель, — смена направления движения.
3. Если найден нулевой концевик, съезжаем с него в направлении съезда с нулевого концевика.
4. В зависимости от используемых приводов может происходить торможение до нулевой скорости для перехода в режим поиска 0-метки.
5. Ищем 0-метку в направлении съезда с нулевого концевика.
6. Если 0-метка найдена, ось тормозится до 0-й скорости.
7. Прибавляем к координате 0-метки смещение нуля станка (параметр **N5n15**) и перемещаемся в новый ноль станка.
8. Задаём новую позицию нуля станка (параметр **N5n16**).
9. Включаем программные концевики (**N5n11**, **N5n10**), если задан соответствующий параметр (**N5n03**).



2.14.3 Настройка безопасного подвода осей

Безопасный подвод осей необходим при выполнении управляющей программы (УП) с заданного кадра или при продолжении исполнения после перемещений в ручном режиме в приостановленной программе. Безопасный подвод определяет простую очерёдность осей с заданными позициями, по которым необходимо осуществлять перемещения перед продолжением управляющей программы. За безопасный подвод отвечают параметры **N3500...N3516** для первого канала и **N3550...N3566** — для второго. Для каждой очерёдности указывается название оси и позиция, в которую по ней необходимо выйти.

Например, если станок фрезерный 3-осевой, то подвод обычно осуществляется по оси Z. В таком случае в параметре N3500 указывают «Z», а в параметре N3501 — безопасную позицию по Z, например, 0 (может соответствовать нулю станка). При начале УП со строки будет осуществлено перемещение по оси Z в безопасную позицию, затем произойдёт перемещение в координаты конца кадра по X и Y, затем в координаты по программе опустится ось Z, и программа продолжит исполнение. Аналогично, если в паузе программы было произведено перемещение по осям, при продолжении УП сначала поднимется ось Z, затем произойдёт перемещение в последнюю позицию программы по X и Y, опустится по Z, и программа продолжит исполнение.

2.14.4 Настройка подготовки исполнения со строки

При использовании функции продолжения УП с определённой строки происходит предварительная подготовка технологических команд и их исполнение. Программа просматривается до указанной строки, и происходит запоминание найденных вызовов M-, S- и T-команд. M-команды группируются согласно параметрам N3660...N3699. В каждой группе M-команды являются взаимоисключающими, и при исполнении будет вызываться только последняя найденная M-команда из группы. Вызовы T- и S-команд происходят вместе с вызовами ассоциированных с ними M-команд. Номера групп с ассоциированными M-командами указываются в параметрах N3650 и N3651 соответственно.

Например, если в параметре N3650 будет указана группа 1, а в параметре N3660 (1-я группа) будет прописано число 6, то вместе с найденным T будет исполнена найденная M6. Если в параметре N3661 (2-я группа) будет указано 3:4:5, а в параметре N3651 — 2, то вместе с S-командой будет исполнена последняя найденная команда из числа: M3, M4 и M5.

Руководство по сопряжению устройства ЧПУ со станками.

Данное руководство предназначено для обучения сопряжению устройств ЧПУ со станками. Предлагаемые ниже порядок работы и стиль программирования не является единственно верным, но может служить примером для подражания при недостатке времени на создание собственного стиля. Для работы необходимы:

- принципиальные схемы электрооборудования станка;
- паспортные данные станка;
- алгоритм работы узлов станка;
- исправный станок с проверенным по возможности электрооборудованием;
- подключенное к станку устройство CNC11 с правильно установленными системными параметрами.

Работа по сопряжению сводится к написанию и отладке программы электроавтоматики и состоит из следующих этапов.

1. Создание нового файла электроавтоматики.
2. Обеспечение работы смазки.
3. Реализация движения по осям.
4. Обеспечение пуска и останова программы.
5. Обеспечение исполнения команд с пульта станка в режиме MDI.
6. Обеспечение исполнения функций M, S и T.



ВНИМАНИЕ!

В процессе отладки электроавтоматики все блокировки станка, осуществляемые УЧПУ, могут быть сняты, поэтому все действия по наладке станка следует осуществлять очень внимательно и осторожно!

3.1 Создание нового файла электроавтоматики

Для создания нового файла ЭА необходимо воспользоваться встроенным редактором (рисунок 3.1.1), расположенным на вкладке **Диагностика > Программа ЭА**. В окне редактора нажимаем кнопку **Файл** и выбираем один из пунктов выпадающего меню **Открыть** или **Открыть в новой** (рисунок 3.1.2). Откроется вкладка выбора проектов **Выбор PLC** (рисунок 3.1.3). Нажмите на кнопку бокового меню **Создать** и выберите в выпадающем меню пункт **Проект** (рисунок 3.1.4). На экране появится диалоговое окно для ввода **уникального имени** ПЛК-модуля (рисунок 3.1.5). По нажатию кнопки ОК или клавиши **Enter** будет создан новая директория с указанным именем, содержащая файлы проекта.

Для запуска проекта необходимо скомпилировать его, нажав кнопку **Скомпилировать**, и перезапустить ПЛК-модуль, нажав кнопку **Перезапустить PLC**.

ВАЖНО: Для выбора модуля ПЛК его имя (имя папки) необходимо прописать в параметре N107!

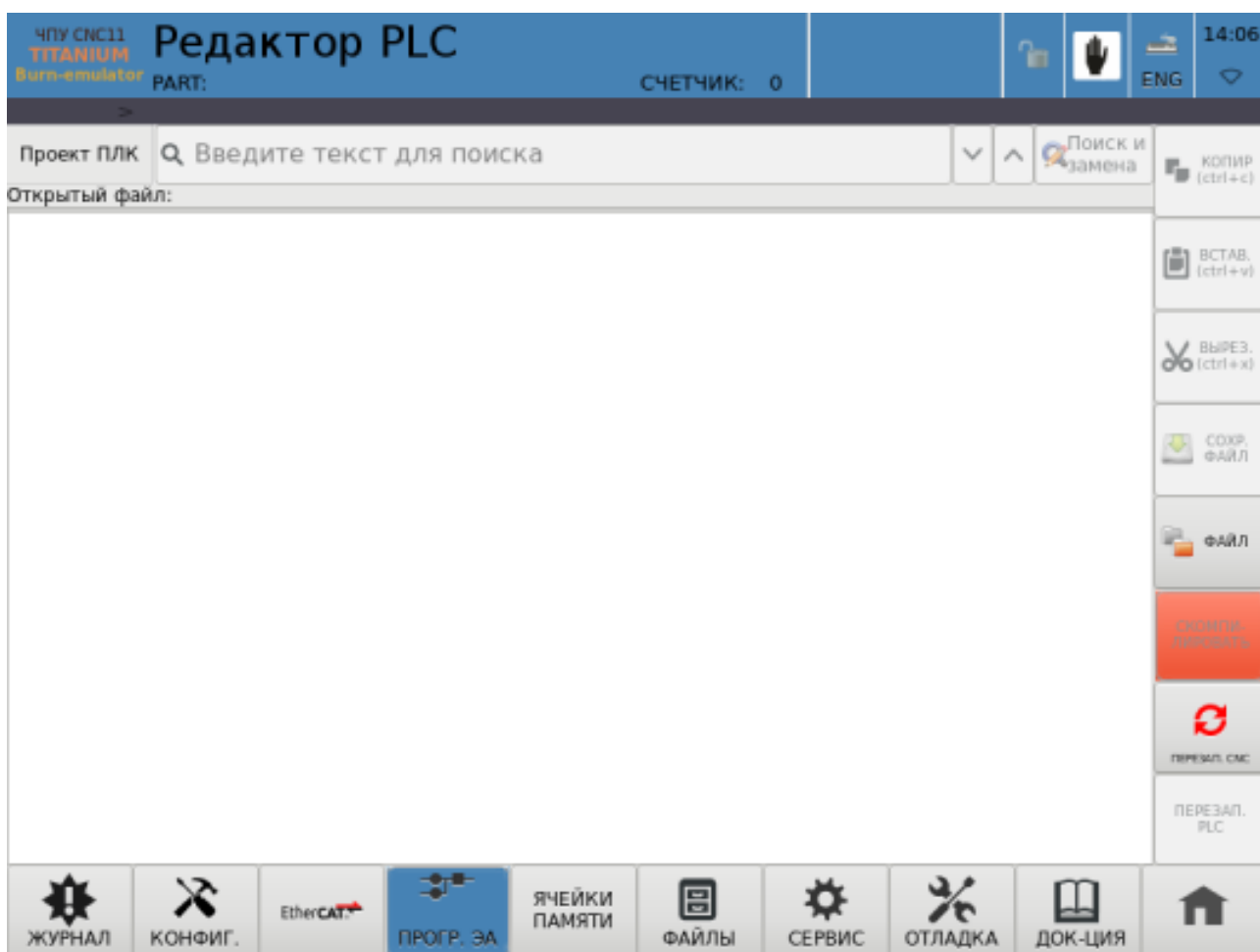


Рис. 3.1.1: Окно редактора ПЛК

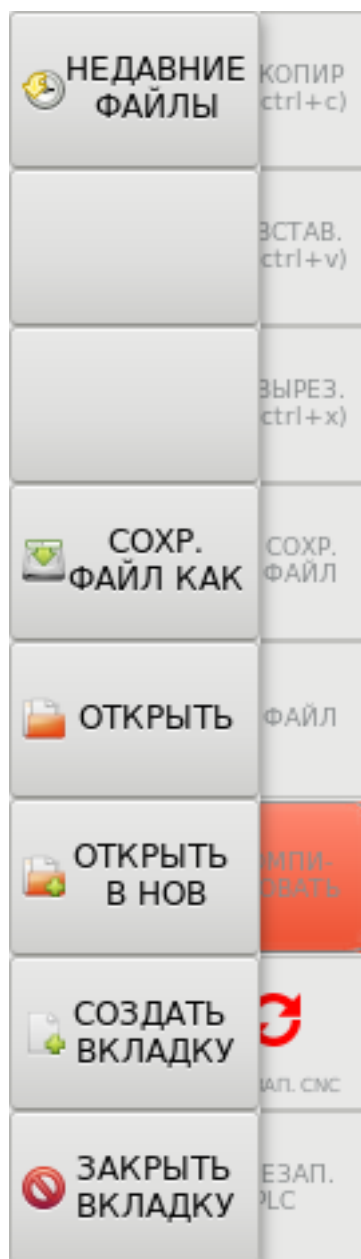


Рис. 3.1.2: Выпадающее меню Файл

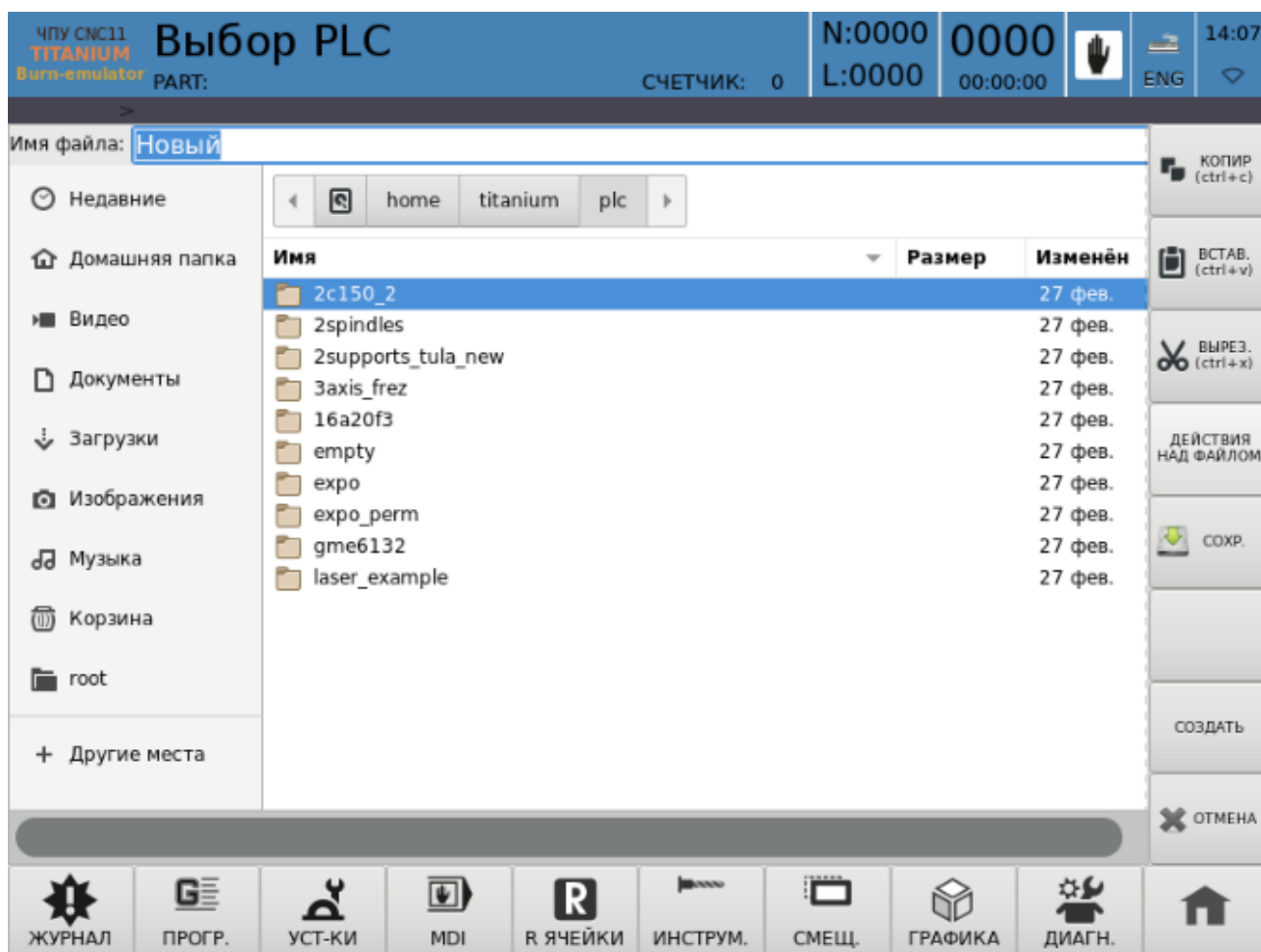


Рис. 3.1.3: Окно выбора проекта ПЛК

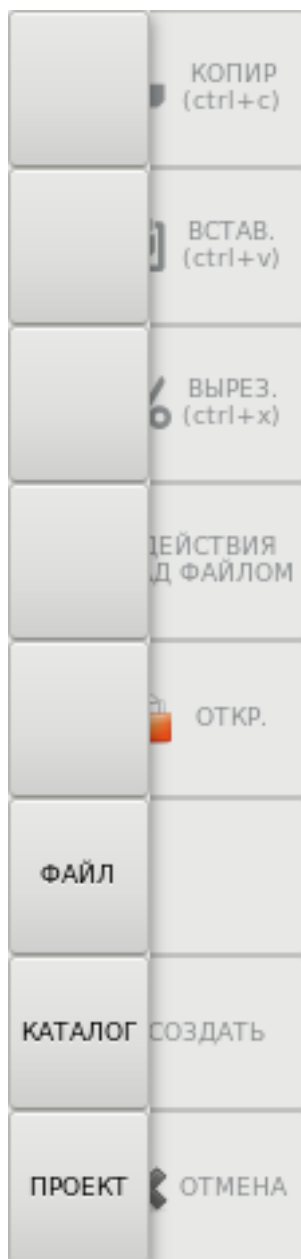


Рис. 3.1.4: Выпадающее меню создания ПЛК

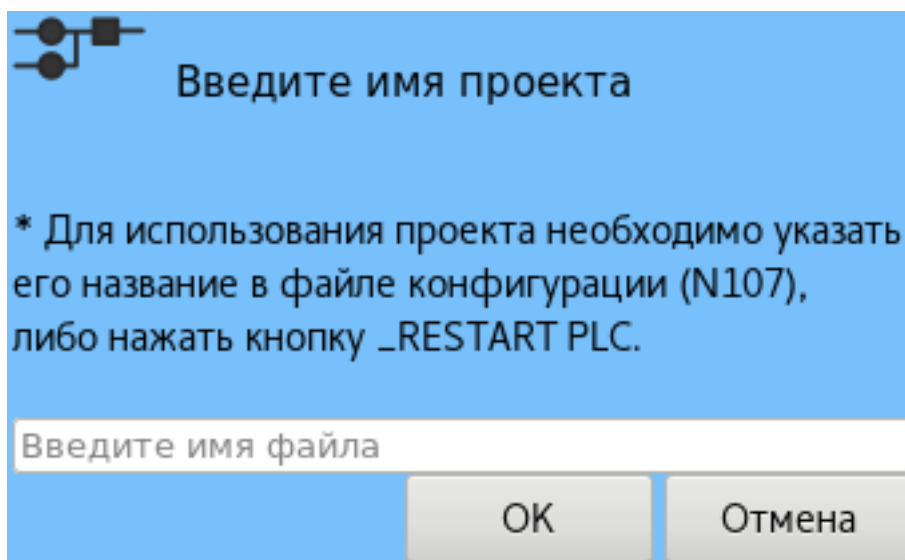


Рис. 3.1.5: Диалог создания ПЛК

3.2 Включение станка

Включение станка можно производить только после появления сигнала **MACH_UI_RDY_I** от ЧПУ. Этот сигнал означает, что ЧПУ полностью запустилась и проинициализировалась. Попытка включения до выдачи этого сигнала возможна, но крайне не рекомендуется, т. к. оператор не будет видеть состояние станка на экране, т. к. графическая подсистема ещё не успела загрузиться.

Для включения станка необходимо подать сигналы включения электроавтоматики станка (включить соответствующие реле) через U-ячейки и проинформировать ЧПУ о том, что станок включен (установить **MACH_ON_O** в 1).

```
1 PLC_TITLE_START
2 #define SB_STANOK_ON I203_5 // Кнопка включения станка (на
   пульте оператора)
3 #define HL_STANOK_ON U205_7 // Светодиод кнопки включения
   станка (на пульте оператора)
4 #define SB_STANOK_OFF I203_6 // Кнопка выключения станка (на
   пульте оператора)
5 #define HL_STANOK_OFF U206_7 // Светодиод кнопки выключения
   станка (на пульте оператора)
6 #define Error_Stanok_On M1_1 // Память ошибки включения станка
7 #define Stanok_On M45_3 // Флаг включения станка
8 #define SF_Stanok_On I1_1 // Вход подтверждения включения
   станка
9 #define KV_Stanok_On U1_1 // Реле включения станка
10 PLC_TITLE_END
```



```
11
12 PLC_VAR_INIT_START
13 T1 = 10; // Время до начала контроля ошибки включения станка (1
    сек)
14 MACH_NO_ERR_0 = true; // Нет аварий (инициализация сделана для
    упрощения примера)
15 PLC_VAR_INIT_END
16
17 PLC_SLOW_START
18 // Алгоритм формирования сигнала включения станка
19 Stanok_On = (SB_STANOK_ON || (Stanok_On && SF_Stanok_On))
20     && !(SB_STANOK_OFF && !CH1_PRG_RUNNIG_I) // Не нажмем
    кнопку выключения
21     && MACH_NO_ERR_0 // Нет аварии
22     && MACH_UI_RDY_I // Есть
    ГОТОВНОСТЬ ЧПУ
23     && !Error_Stanok_On; // Пока нет ошибки
24 KV_Stanok_On = Stanok_On; // Реле включения станка
25 MACH_ON_0 = Stanok_On; // Сообщить ЧПУ, что
    станок включен
26 TS1(Stanok_On); // Включение таймер контроля ошибки
27 HL_STANOK_ON = SB_STANOK_ON || Stanok_On; // Включение диода
    по кнопке или сигналу
28 HL_STANOK_OFF = SB_STANOK_OFF || !Stanok_On; // Включение диода
    по кнопке или сигналу
29 // Ответ включения не пришел - формируем ошибку!
30 Error_Stanok_On = ((TOS1 && !SF_Stanok_On)
31 || Error_Stanok_On) && !MACH_UI_ESC_IO; // На самопитание, пока
    не сбросим
32 ERR(Error_Stanok_On, 1, "Ошибка Включения станка"); // Показать
    ошибку
33 PLC_SLOW_END
```

3.3 Включение приводов

Включение приводов осуществляется только, если станок включен. В общем случае, необходимо подать сигналы включения приводов и выставить сигнал **MACH_DRV_ON_0**, который информирует ЧПУ о том, что по всем приводам можно осуществлять движение. Если MACH_DRV_ON_0 выставлен в 0, ЧПУ не будет выполнять команд движения по осям и, соответственно, не будет исполнять УП.

Алгоритм включения каждого привода в отдельности будет варьироваться от станка к станку. В простых случаях достаточно:

- сбросить ошибки pid-регулятора (**AX1_PID_ERR_IO** установить в 0) и рассогласования (**AX1_PID_RESET_0** установить в 1);



- подать сигнал включения привода через U-ячейки;
- выставить **AX1_DRV_OUT_O** в 1 для разрешения выдачи управляющего задания на привод;
- включить слежение по оси (записать 1 в **AX1_PID_ON_O**).

Если алгоритм реализован правильно, а ось настроена, после включения приводов ось должна войти в слежение (удержание текущей позиции).

3.4 Сигналы подготовки перемещения

```
1 PLC_TITLE_START
2 #define SB_DRIVES_ON      I204_5 // Кнопка включения приводов
   (на пульте оператора)
3 #define HL_DRIVES_ON      U213_7 // Светодиод кнопки включения
   приводов (на пульте оператора)
4 #define SB_DRIVES_OFF     I204_6 // Кнопка выключения приводов
   (на пульте оператора)
5 #define HL_DRIVES_OFF     U214_7 // Светодиод кнопки выключения
   приводов (на пульте оператора)
6 #define Error_Driver_X    M3_1  // Ошибка привода X
7 #define Driver_On         M46_1 // Привода включены
8 #define i_Ready_Driver_X  I2_1  // Готовность привода X
9 #define Error_Driver_X    M2_1  // Ошибка рассогласования
   привода X
10 #define Error_Driver      M1_2  // Ошибка рассогласования
   приводов
11 #define KV_0n_X           U2_4   // Сигнал на включение привода
   X
12 #define SB_RESET          I207_5 // Кнопка сброса (на пульте
   оператора)
13 #define HL_RESET          U237_7 // Светодиод кнопки сброса (на
   пульте оператора)
14 PLC_TITLE_END
15
16 PLC_VAR_INIT_START
17 T7 = 20; // Время задержки включения приводов для контроля
   ошибки
18 PLC_VAR_INIT_END
19
20 PLC_SLOW_START
21 // Сигналы подготовки перемещения (деблокировка приводов подач и
   шпинделя)
22 TS7(Driver_On); // Не контролируем ошибку, пока этот таймер
   (T0S7) не включен (не сработает реле)
23 // Контроль ошибки
```



```
24 Error_Driver_X = ((TOS7 && !i_Ready_Driver_X) || Error_Driver_X)
    && !MACH_UI_ESC_IO;
25 ERR(Error_Driver_X,4, "Нет готовности привода -X- "); //
    Показываем ошибку привода X
26 // Сигнал включения приводов
27 Driver_On = (SB_DRIVES_ON || Driver_On) // Самопитание по кнопке
    && !(SB_DRIVES_OFF && !CH1_PRG_RUNNIG_I) //
    Выключение по нажатию на кнопку
29 && MACH_UI_RDY_I && MACH_NO_ERR_0 // Если ЧПУ готово
    и нет аварии
30 && Stanok_On // Если станок включен
31 && !Error_Driver // Нет ошибки драйверов
32 && !MACH_DRV_ERR_IO; // Не превышено
    рассогласование
33
34 KV_On_X = i_Ready_Driver_X; // Включаем привод
35 AX1_MOVE_ON_0 = MACH_DRIFT_ON_0 = Driver_On; // По включению
    приводов - коррекция дрейфа и разрешение перемещения по оси
36 AX1_PID_RESET_0 = HL_RESET = SB_RESET; // Сброс рассогласования
    по кнопке сброса и зажечь диод на ней
37 MACH_DRV_ERR_IO = MACH_DRV_ERR_IO && !SB_RESET && !
    CH1_PRG_RUNNIG_I; // Сброс
    рассогласования
38 // Зажжем диоды кнопок включения/выключения приводов
39 HL_DRIVES_ON = Driver_On || SB_DRIVES_ON;
40 HL_DRIVES_OFF = !Driver_On || SB_DRIVES_OFF;
41 PLC_SLOW_END
```

3.5 Выбор режима работы

```
1 PLC_TITLE_START
2 // Кнопка, диод автоматического режима
3 #define SB_AUTO_MODE I203_1
4 #define HL_AUTO_MODE U201_7
5 // Кнопка, диод покадрового режима
6 #define SB_STEP_MODE I203_2
7 #define HL_STEP_MODE U202_7
8 // Кнопка, диод ручного режима
9 #define SB_MANU_MODE I203_3
10 #define HL_MANU_MODE U203_7
11 // Кнопка, диод режима выхода в 0
12 #define SB_HOME_MODE I203_4
13 #define HL_HOME_MODE U204_7
```



```
14 // Кнопка, диод режима преднабора
15 #define SB_MDI I204_1
16 #define HL_MDI U209_7
17 PLC_TITLE_END
18
19 PLC_SLOW_START
20 // ВЫБОР РЕЖИМА РАБОТЫ
21 RI200 = (SB_MANU_MODE || CH1_UI_MAN_IO || SB_X_1 || SB_X_10 ||
          SB_X_100
          || SB_FEED_1 || SB_FEED_2 || SB_FEED_3) &&
          ((CH1_PRG_RUNNIG_I && CH1_PRG_STOP_I) || !(
          CH1_PRG_RUNNIG_I));
24 RI201 = (SB_AUTO_MODE || CH1_UI_AUTO_IO) && !(CH1_PRG_RUNNIG_I
          && CH1_MDI_I);
25 RI202 = SB_STEP_MODE || CH1_UI_STEP_IO;
26 RI203 = SB_HOME_MODE || CH1_UI_HOME_IO;
27 RI204 = (SB_MDI || CH1_UI_MDI_IO) && !(CH1_PRG_RUNNIG_I &&
          CH1_AUTO_I);
28 // Запросы установки режимов из интерфейса, после чтения -
          СБРОСИТЬ!
29 CH1_UI_MAN_IO = 0;
30 CH1_UI_HOME_IO = 0;
31 CH1_UI_MDI_IO = 0;
32 CH1_UI_AUTO_IO = 0;
33 CH1_UI_STEP_IO = 0;
34 // Автоматический режим
35 CH1_AUTO_0 = (R201 || R202 || R204 || CH1_AUTO_I || CH1_AUTO_0)
          && !R200 && !R203;
36 // Покадровый режим (автомат)
37 CH1_STEP_0 = (R202 || CH1_STEP_I || CH1_STEP_0) && !R200 && !
          R201 && !R203 && !R204;
38 // Ручной режим управления
39 CH1_MAN_0 = (R200 || R203 || CH1_MAN_I || CH1_MAN_0) && !R201 &&
          !R202 && !R204;
40 // Выход в ноль
41 CH1_HOME_0 = (R203 || CH1_HOME_I || CH1_HOME_0) && !R200 && !
          R201 && !R202 && !R204;
42 // Преднабор (MDI)
43 CH1_MDI_0 = (R204 || CH1_MDI_I || CH1_MDI_0) && !R200 && !R203
          && !R201;
44 // Установка интерфейса преднабора при выборе режима МДИ
45
46 // Подсветим диоды режимов
```



```
47 HL_AUTO_MODE = CH1_AUTO_I && !CH1_STEP_I && !CH1_MDI_I;  
48 HL_MANU_MODE = CH1_MAN_I && !CH1_HOME_I;  
49 HL_STEP_MODE = CH1_STEP_I;  
50 HL_HOME_MODE = CH1_HOME_I;  
51 HL_MDI = CH1_MDI_I;  
52 PLC_SLOW_END
```

3.6 Настройка приводов

3.6.1 Алгоритм базовой настройки привода

1. Определить и установить значения делителя (N5n13) и множителя (N5n14) по оси.
2. Установить медленный разгон и торможение по оси (небольшие значения для параметров N6n00, N6n01)
3. Подобрать такие значения для пропорционального коэффициента (N6n21) и добротности (N6n08), чтобы привод держал свою позицию, в то время как пропорциональный коэффициент должен быть близок к нулю.
4. Повысить значение коэффициента управления по скорости (N6n09) примерно до значения, близкого к 1.0. Значение 1.0 (или выше) брать нельзя, т. к. это вызовет перерегулирование привода из-за добавочного управления от ПИД-регулятора.
5. Повышать добротность до тех пор, пока рассогласование не перестанет расти в процессе движения (в идеальном случае оно должно быть примерно равно нулю).
6. Снизить коэффициент управления по скорости и повысить пропорциональный коэффициент для задания необходимой жёсткости привода. Коэффициент управления по скорости можно установить приблизительно в 0.5, что обеспечит оптимальную жёсткость.
7. Подобрать ускорение, при котором не будет перерегулирования. Ускорение, превышающее характеристики привода, будет вызывать перерегулирование при разгоне и торможении.
8. В случае неудовлетворительного качества движения варьировать пропорциональный коэффициент, коэффициент управления по скорости и ускорение.



3.7 Пуск управляющей программы

```
1 PLC_TITLE_START
2 // Диоды/кнопки пуска/стопа программы
3 #define SB_START_PRG I208_5
4 #define HL_START_PRG U245_7
5 #define SB_STOP_PRG I208_6
6 #define HL_STOP_PRG U246_7
7 // Кнопка сброса
8 #define SB_RESET I207_5
9
10 #define PUSK_EN M100_1
11 PLC_TITLE_END
12
13 PLC_SLOW_START
14 // Флаг разрешения пуска программы
15 PUSK_EN = ((CH1_PRG_CHOSEN_I && CH1_AUTO_I) || CH1_MDI_I)
16           && ошибка!<зажимапатрона__>
17           && станок<вкл_> && привода<вкл_>;
18
19 RI253 = SB_START_PRG || CH1_UI_PRG_RUN_IO ; // Пуск программы
20 CH1_UI_PRG_RUN_IO = 0; // Запрос на пуск УП, сбрасываем после
   чтения!
21 CH1_PRG_RUN_IO = PUSK_EN && R253 && !CH1_PRG_RUNNIG_I; // Пуск УП
22
23 // Стоп УП
24 RI254 = (SB_STOP_PRG || CH1_UI_PRG_STOP_IO || (CH1_M_I && ((
   CH1_M_R == 30) || (CH1_M_R == 2)))) && CH1_PRG_RUNNIG_I;
25 C100_I = R254; // Считаем количество нажатий на кнопку
26 CH1_UI_PRG_STOP_IO = 0; // Запрос на стоп УП, сбрасываем после
   чтения
27 CH1_PRG_STOP_IO = ((C100 == 1) || CH1_PRG_STOP_IO || (CH1_M_I &&
   (CH1_M_R == 0))) && !R253; // СТОП, если нажали один раз,
   или по ячейкам...
28
29 // СБРОС, если нажали два раза, или по рассогласованию или по
   кнопке сброса...
30 CH1_PRG_ABRT_0 = (C100 == 2) || MACH_DRV_ERR_IO || SB_RESET;
31 C100_RESET = R253 || ((C100 == 2) && !SB_STOP_PRG); // Сбросим
   счетчик
32 // Формируем сообщения
33 VIEW(CH1_PRG_RUN_IO, 10, "Команда на пуск программы, канал № 1")
   ;
```



```
34 VIEW(CH1_PRG_STOP_I0, 11, "Команда на останов программы, канал  
    № 1");  
35 VIEW(CH1_PRG_ABRT_0, 12, "Команда на сброс программы, канал № 1"  
    );  
36 VIEW(!PUSK_EN && R253), 13, "Пуск программы невозможен, канал  
    № 1");  
37 // Зажжем диоды  
38 HL_START_PRG = ((CH1_PRG_RUNNIG_I && !CH1_PRG_STOP_I && (C100 ==  
    0)) || SB_START_PRG);  
39 HL_STOP_PRG = (CH1_PRG_STOP_I && T0S1) || SB_STOP_PRG || (C100  
    == 2);  
40 PLC_SLOW_END
```

3.8 Ручной режим

Ниже приведен код, описывающий работу ручного режима для трёхосевой конфигурации.

ВАЖНО: для задания скорости для осей и шпинделя необходимо использовать макрос `SPEED_TO_RT!`

3.9 Смена инструмента

```
1 RI19 = CH1_T_I && <SMENA_ZAKONCHENA>;  
2 CH1_T_0 = SELECT_TOOL_AND_OFFSETS(R19, 1, CH1_T_N_NEW_R,  
    CH1_T_B_NEW_R);  
3 if (R19) { RI19 = 0; } // Для подряд идущих Т-команд  
4 VIEW(CH1_T_ID_W < 0, 39, "Инструмент в таблице отсутствует.  
    Канал №1.");  
5 VIEW(CH1_T_OFFSETS_ID_W < 0, 40, "Инструмент в таблице  
    отсутствует. Канал №1.");
```

Функция `SELECT_TOOL_AND_OFFSETS` ищет инструмент в таблице инструментов по номеру инструмента и привязки, затем выбирает его. Сигнал `<SMENA_ZAKONCHENA>` формируется, исходя из особенностей механической части станка.



3.10 Управление шпинделем

3.10.1 Включение шпинделя

```
1 PLC_TITLE_START
2 #define Shpind_On    U1_4 // Выход на реле включения шпинделя
3 #define usl_on_shpind    M43_2 // Флаг включения шпинделя
4 #define usl_off_shpind    M43_3 // Флаг выключения шпинделя
5 #define Kod_M    M9_1 // Флаг поступления М-кода
6
7 #define M3    M10_1 // Шпиндель по часовой стрелке
8 #define M4    M10_2 // Шпиндель против часовой стрелки
9 #define M5    M10_3 // Шпиндель стоп!
10
11 #define polzuch    M43_4 // Скорость смены передачи (если
    нужно)
12 #define nor_shpind    M43_5 // Нормальный режим шпинделя
13 #define Tolchok    M42_7 // Толчок шпинделя
14 PLC_TITLE_END
15
16 PLC_VAR_INIT_START
17 T6 = 5; // Задержка на деблокировку привода после снятия
    задания
18 T5 = 5; // Задержка на выдачу задания на S после деблокировки
19 PLC_VAR_INIT_END
20
21
22 PLC_FAST_START
23 // Выдача задания на ЦАП шпинделя
24 CH1_SP_VAL_W = SPEED_T0_RT(((polzuch && !Tolchok ) || Tolchok) *
    P6 * ((-1) * (Tolchok && SB_SPINDLE_CCW)
25 + 1 * (Tolchok & SB_SPINDLE_CW) + 1 * (Tolchok & SB_TOLCHOK) + (
    nor_shpind) * (M4 + ((!M4) * (-1))));
26 PLC_FAST_END
27
28
29
30 PLC_SLOW_START
31 // Включаем шпиндель
32 Shpind_On = (((((usl_on_shpind || Shpind_On) & !TOS6) & !Tolchok
    )
33 || (Tolchok && (SB_SPINDLE_CCW || SB_SPINDLE_CW)
```



```
&& M5)))  
34         && !Error_Shpind  
35         && !Error_Zazh_Patr  
36         && Driver_On  
37         && Stanok_On;  
38  
39 Kod_M = CH1_M_I && CH1_AUTO_I; // Поступил М-код  
40  
41 // Формируем бит шпинделя  
42 M3 = ((Kod_M && ((CH1_M_R == 3) || (CH1_M_R == 13)))  
43       || (CH1_MAN_I && SB_SPINDLE_CW && !Tolchok)  
44       || M3)  
45       && !(Kod_M && ((CH1_M_R == 2) || (CH1_M_R == 5) || (CH1_M_R  
46       == 30)  
47       || (CH1_M_R == 4) || (CH1_M_R == 14) || (CH1_M_R == 19) || (  
48       CH1_M_R == 45)))  
49       && !(CH1_MAN_I && (SB_SPINDLE_CCW || SB_SPINDLE_OFF))  
50       && !(CH1_AUTO_I && SB_SPINDLE_OFF)  
51       && !Error_Shpind  
52       && Stanok_On && Driver_On;  
53  
54 // Формируем бит выключения шпинделя  
55 M5 = ((Kod_M && ((CH1_M_R == 2) || (CH1_M_R == 5) || (CH1_M_R ==  
56       30)))  
57       || (CH1_MAN_I && SB_SPINDLE_OFF && !Tolchok)  
58       || (CH1_AUTO_I && SB_SPINDLE_OFF)  
59       || Tolchok || !Zazh_Patr || !MACH_NO_ERR_0 || !Stanok_On  
60       || !Driver_On  
61       || Error_Smaz_S || Error_Shpind || M5)  
62       && !(Kod_M && ((CH1_M_R == 3) || (CH1_M_R == 13) || (CH1_M_R ==  
63       4) || (CH1_M_R == 14)  
64       || (CH1_M_R == 19) || (CH1_M_R == 45)) && !Cikl_smpered  
65       )  
66       && !(CH1_MAN_I && (SB_SPINDLE_CW || SB_SPINDLE_CCW) && !  
67       Tolchok && Zazh_Patr)  
68       && !(CH1_AUTO_I && ((SB_SPINDLE_CW && M3_shtrih) | (  
69       SB_SPINDLE_CCW && M4_shtrih)) && Zazh_Patr);  
70  
71 usl_on_shpind = ((M3 || M4) && !M5) && !usl_off_shpind; //  
72     Условие включения шпинделя  
73 usl_off_shpind = (M5 && !(M3 || M4)) && Shpind_On; // Условие  
74     выключения шпинделя  
75
```



```
66 TS5(usl_on_shpind); // Задержка на деблокировку привода после
    снятия задания
67 TS6(usl_off_shpind); // Задержка на выдачу задания на S после
    деблокировки
68 nor_shpind = ((TOS5 && (M3 || M4)) || nor_shpind) && !
    usl_off_shpind && !M5 && MACH_NO_ERR_0 && Stanok_On; //
    Шпиндель в нормальном режиме
69
70 // Подсветим диоды шпинделя ПО, ПРОТИВ, ВЫКЛ
71 HL_SPINDLE_CW = SB_SPINDLE_CW || (Shpind_On && (CH1_SP_VAL_W >
    0));
72 HL_SPINDLE_CCW = SB_SPINDLE_CCW || (Shpind_On && (CH1_SP_VAL_W <
    0));
73 HL_SPINDLE_OFF = SB_SPINDLE_OFF || (!Shpind_On && (CH1_SP_VAL_W
    == 0));
74
75 // Ответ по M
76 M_shpind = ((CH1_M_R == 3) && M3 && nor_shpind && !polzuch &&
    Shpind_On)
77     || ((CH1_M_R == 4) && M4 && nor_shpind && !polzuch &&
    Shpind_On)
78     || ((CH1_M_R == 5) && M5 && !nor_shpind && !polzuch
    && !Shpind_On);
79 CH1_M_0 = CH1_M_I && ((CH1_M_R == 0) || (CH1_M_R == 1) || (((
    CH1_M_R == 30) || (CH1_M_R == 2))
80     && M5 && !M3 && !M4 && !nor_shpind && !polzuch) ||
    M_shpind);
81
82 PLC_SLOW_END
```

3.10.2 Реверс шпинделя

Реверс используется для цикла нарезания резьбы метчиком. Номер M-функции может быть изменен.

```
1 PLC_TITLE_START
2 #define M34      M10_7 // M-функция реверса шпинделя
3 #define M34_0k  M10_8 // Ответ M-функции реверса шпинделя
4 #define M35      M11_1 // Отмена реверса шпинделя
5 PLC_TITLE_END
6
7 PLC_FAST_START
```



```
8 // Вычисление задания на шпиндель
9 CH1_SP_VAL_W = k * SPEED_TO_RT(((P6 * (Slow_Spindle_CW +
    Slow_Spindle_CCW * (-1))))
10 // Вычисление знака задания на шпиндель
11     +!M5 * (nor_shpind * kod * (M3 + M4 * (-1)) * (!M34 + (
    M34&&M34_0k) * (-1))
12         * ((Transmiss == 1) * P3/1000.0
13           + (Transmiss == 2) * P4/1000.0
14           + (Transmiss == 3) * P5/1000.0)))));
15 PLC_FAST_END
16
17 PLC_SLOW_START
18 // М-функция реверса шпинделя
19 M34 = ((Kod_M && (CH1_M_R == 34)) || M34)
20     && !(Kod_M && (CH1_M_R == 3))
21     && !(Kod_M && (CH1_M_R == 4))
22     && !(Kod_M && (CH1_M_R == 35))
23     && !M_Shpind_Stop
24     && !SB_RESET;
25 // Ответ по М-функции реверса шпинделя
26 M34_0k = M34 && (Shpind_Speed_Zero || M34_0k);
27 // Определение минимальной скорости шпинделя для начала его
    реверса
28 Shpind_Speed_Zero = (AX4_RPM_AVG_R > -10.0) && (AX4_RPM_AVG_R <
    10.0);
29 // Ответ по М-функции шпинделя
30 M_shpind=((CH1_M_R == 3) && M3 && Shpind_On && TOS33 && !n_zero)
31     ||((CH1_M_R == 4) && M4 && Shpind_On && TOS33 && !
    n_zero)
32     ||((CH1_M_R == 5) && M5 && !Shpind_On && n_zero )
33     ||((CH1_M_R == 34) && M34 && M34_0k)
34     ||((CH1_M_R == 35) && !M34 )
35     ||((CH1_M_R == 19) && M19_0k);
36 // Ответ по М-функциям
37 CH1_M_0 = SB_RESET || (CH1_M_I && ((CH1_M_R == 0)
38     ||(CH1_M_R == 1)
39     ||(((CH1_M_R == 30) || (CH1_M_R == 2)) && ((!
    Shpind_On && M95) || M96))
40     ||M_shpind
41     ||M_0s_C
42     ||M4x_ok
43     ||M_ok
44     ||M_SOZH
```



```
45         ||M_Patr_0k));  
46 PLC_SLOW_END
```

3.10.3 Конфигурации с несколькими шпинделями

Внимание! На данный момент поддержка конфигураций станков с несколькими шпинделями запущена в тестовом режиме. Это означает, что в будущих версиях ЧПУ принципы управления шпинделями могут измениться, а описание параметров, связанных с настройкой нескольких шпинделей, могут отсутствовать.

Если в системе используется более одного шпинделя, то по S-команде в ПЛК поступают следующие данные.

1. В CH1_SP_N_NEW_R записывается порядковый номер предварительно выбранного в программе G-кодов шпинделя.
2. В CH1_S_RPM_NEW_R записывается новое задание (обороты) для выбранного шпинделя.
3. В CH1_S_I записывается 1.

При записи из ПЛК в ячейки CH1_SP_VAL_W, CH1_UI_S_RPM_W, CH1_SP_GEAR_W, CH1_SP_BY_PLC_O, CH1_SP_IS_AX_IO изменения применяются для текущего шпинделя, номер которого указан в ячейке CH1_SP_N_NEW_R. Данная модель позволяет реализовать поочерёдное управление шпинделями.

Для одновременного управления несколькими шпинделями рекомендуется записывать данные в соответствующие именованные ячейки.

3.10.4 Подчинённые оси

Подчинённые оси могут использоваться в порталных станках, где две оси должны синхронно осуществлять передвижение портала. У оси, которая может выступать в роли подчинённой необходимо выставить параметр **N5n42** в 1, а в параметр **N5n43** занести номер главной оси, с которой будет осуществляться синхронизация. В параметр **N5n44** необходимо занести максимально возможное значение рассинхронизации позиции осей по датчику обратной связи. По превышению данного значения будет выдан сигнал ошибки.

Режим подчинённой оси включается флагом **AX{номер}_MODE_SLAVE_O** в модуле ПЛК. Главная ось начнёт движение только после того, как будут выставлены ячейки разрешения движения **AX{номер}_MOVE_ON_O** и у главной оси, и у подчинённой оси. В режиме подчинённой оси значение ячейки **AX{номер}_MOVE_ON_I** запроса на движение будет дублировать значение соответствующей ячейки главной



оси. После включения режима подчинённой оси подчинённая ось полностью дублирует движение главной оси и имеет ту же абсолютную координату. В этом режиме для подчинённой оси применяются ускорения главной оси, а также учитываются её концевые и программные ограничители. В случае, если у подчинённой оси допустимые ускорения меньше, чем у главной, может потребоваться занижение ускорений главной оси в параметрах до значений, указанных у подчинённой.

Параллельный выход в 0 с выключенным режимом подчинённой оси может использоваться для начальной синхронизации позиции осей. Однако в этом случае у осей должны быть разные концевики 0-метки, в противном случае возможен выход в 0 только для главной оси с включенной синхронизацией подчинённой оси.

В случае рассинхронизации (согласно N5n44) для обеих осей будет выставлена в 1 ячейка AX\${номер}_PID_ERR_IO, а также будет выставлена ячейка ошибки приводов MACH_DRV_ERR_IO.

4.1 Диагностика датчиков обратной связи

4.1.1 Неисправность инкрементального датчика

Видимые симптомы

При неисправности датчика могут наблюдаться один или несколько из следующих симптомов.

- Физически суппорт не доезжает до заданной позиции, но рассогласование по окончании движения близко к нулю.
- Искажения при изготовлении деталей либо «плывущие» размеры.
- Количество импульсов на один оборот датчика не соответствует его заявленной дискретности.
- Периодически ось совершает резкие перемещения («дёргается»), а периодичность совпадает с оборотом датчика.

Описание проблемы

Причиной может служить выход из строя датчика (либо превышение максимально возможной частоты вращения). В результате может происходить потеря импульсов, передаваемых в ЧПУ. Потеря импульсов, в свою очередь, приведёт к искажению текущих координат, что повлечёт за собой изготовление деталей с неправильными («плывущими») размерами.

Если ось совершает резкие перемещения с периодичностью датчика, это может означать, что кодовый диск датчика может быть испорчен (например, механическое повреждение).



Диагностика

Для выявления подобных проблем рекомендуется включать **защиту поиска «0»-метки** (N5n23). Если произошла потеря импульсов, а по оси найдена очередная «0»-метка (например, в процессе выхода в «0» станка), то система выдаст сообщение об ошибке несоответствия дискретности датчика и позиции 0-метки. Данная ошибка может служить средством диагностики неисправности датчика «0»-метки. Под неисправностью понимается как неисправность самого датчика, так и неисправность кабеля, либо разъёма подключения (отсутствие экрана, перебитый кабель, неправильная распайка кабеля, плохой контакт и т. п.).

Для проверки дискретности датчика можно сделать один оборот оси, на которой закреплён датчик, и убедиться, что изменение абсолютного значения датчика соответствует его дискретности. Например, в случае неправильной распайки может наблюдаться уменьшение дискретности на несколько порядков (на один оборот может быть получено лишь несколько импульсов).

Для проверки исправности датчика оси можно использовать следующий алгоритм.

1. Проехать по диагностируемой оси на максимальной скорости в одном направлении.
2. Вернуться в исходную позицию на медленной скорости.
3. Повторить пункты алгоритма №1 и №2 несколько раз.
4. Осуществить выход в «0». Если по выходу в «0» будет выдана ошибка несоответствия дискретности «0»-метки, вероятно, датчик неисправен.

Решение

В качестве решения можно заменить датчик, поменять соединительный кабель, убедиться в отсутствии плохого контакта.

4.2 Диагностика движения по осям

4.2.1 Неверные параметры приводов

Видимые симптомы

Неправильно настроенная ось может привести к появлению следующих видимых проблем.



- При попытке движения ось начинает медленно «ползти» в том же направлении.
- Ось на большой скорости двигается корректно, но при останове ось начинает колебательные движения (при условии отсутствия люфта).
- По окончании движения ось не доезжает до конечной точки, а затем медленно «ползёт» к заданной точке.
- Ось переезжает конечную точку движения, а затем возвращается назад.

Диагностика

- Убедиться, что при перемещении на определённое расстояние ось по факту проезжает такое же расстояние, как и на экране ЧПУ.
- Убедиться, что датчик ОС на один оборот выдаёт в ЧПУ правильное количество импульсов.
- Убедиться, что нет препятствий движению оси, а также, что нет резких значительных изменений нагрузки на ось.
- Убедиться в отсутствии больших люфтов.

Решение

Необходимо проверить и исправить параметры, отвечающие за настройку движения по оси: N6n00, N6n01, N6n08, N6n09, N6n21 и т. д.