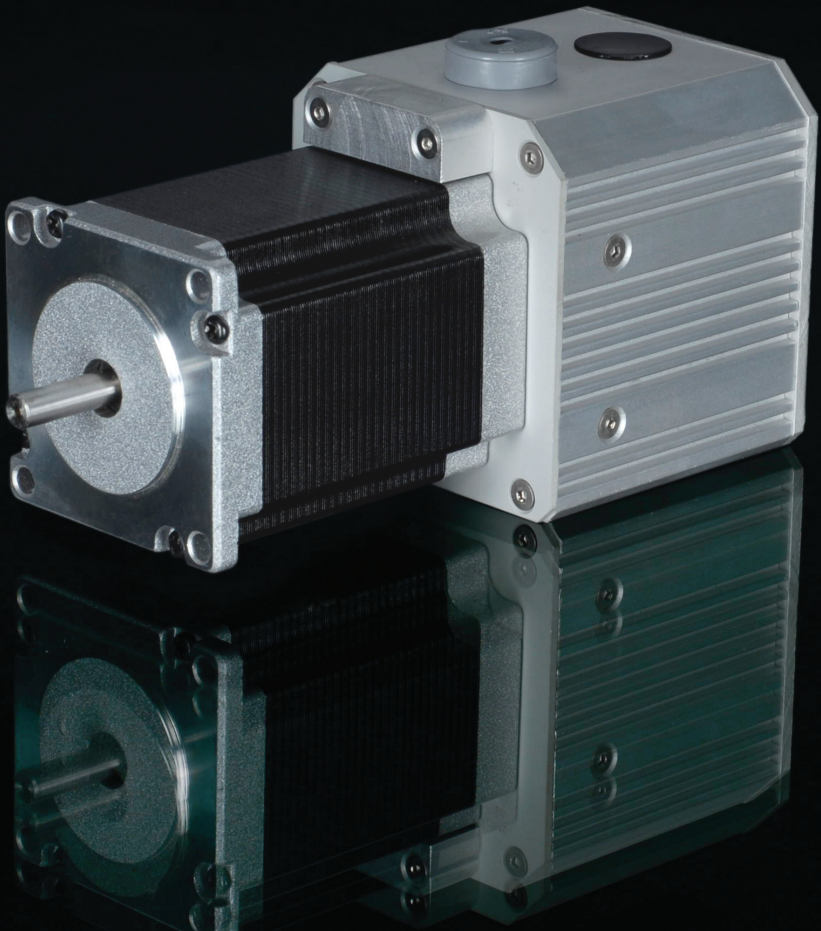


Интегрированный сервопривод СПШ10



III - язык программирования

Интегрированный сервопривод СПШ10

III – язык программирования

Версия 3.0

СПШ10 – это интегрированный сервопривод на базе гибридного шагового электродвигателя, в котором используется бесшаговое (векторное) управление на основе адаптированного специально для шаговых двигателей алгоритма.

Данный документ описывает язык программирования сервопривода СПШ10 и примеры программ для решения типовых задач.

ЗАО «Сервотехника» не возлагает на себя обязанность оповещать пользователей СПШ10 о появлении обновлений комплекта документации и программного обеспечения. Пожалуйста, следите за новостями на сайте компании www.servotechnica.ru.

Содержание

Язык программирования СПШ10	4
Команды	5
События	9
Встроенные переменные и функции	10
Примеры программ	11

Язык программирования СПШ10

В системе управления СПШ10 реализован программируемый логический контроллер. В данной версии СПШ10 в энергонезависимой памяти системы управления может храниться до 8 пользовательских программ. При этом одна из программ, указанных пользователем, может запуститься при включении.

Разработка и отладка программ может выполняться в Мотомастер[®], для этого в него включен текстовый редактор, расположенный в окне «Контроль». Время между циклами обработки программы интерпретатором не превышает 5 мс.

В текущей версии редактора программ Мотомастер[®] существует ряд ограничений:

- не допускаются пустые строки в программе, в т. ч. и последней;
- допускается использование только английской раскладки клавиатуры;
- используйте регистр команд в строгом соответствии с приведенным описанием;
- размер каждой программы (в объектном коде) не может превышать 128 байт. При превышении размера указанного значения компилятор выдаст сообщение об ошибке.

Все константы могут быть введены в десятичном или шестнадцатеричном форматах. Например: P=10 или P=0x0A

Принятые обозначения

X, Y, Z - при написании программ пользователю доступны три переменные общего назначения. X, Y, Z-переменные могут хранить знаковые 32-х битные числа. Данные переменные обнуляются перед запуском программы на выполнение.

var - переменные.

const - целочисленное знаковое число.

op - арифметическая или логическая операция.

Ниже приведен список допустимых арифметических и логических операций.

Операция	Описание
+	Сумма
-	Вычитание
/	Деление
•	Умножение
&	Логическое побитовое «И»
.....	Логическое побитовое «ИЛИ»

Пример использования операций:

P=P+1000

P=P/2

W=X-10

W=Z•2

X=Y&1

<Выражение> - используется в данном руководстве для обозначения выражений, состоящие из переменных и/или констант:

var
const
var op const
var op var

<Условие>

var=const
var>const
var<const
var=var
var>var2
var<var2
var op var2=var3
var op var2>var3
var op var2<var3
var op const 1=const2
var op const 1>const2
var op const 1<const2
var op const=var2
var op const>var2
var op const<var2
PORT.N=<Выражение>(цифровой вход)

В качестве операции в условии не может использоваться • и /.

Команды

Установка режима движения.

Код команды	Описания	Примечания
P= <Выражение>	Установить задание контура позиции (дискрет).	Автоматически замыкается контур положения. Используются текущие настройки контура тока и скорости.
P= <Выражение> W= <Выражение>	Установить задание контура позиции (дискрет), установить максимальную скорость вращения (об/мин).	Аналогично.

Код команды	Описания	Примечания
P=<Выражение> W=<Выражение> A=<Выражение>	Установить задание контура позиции (дискрет), установить максимальную скорость вращения (об/мин), установить текущее ускорение (дискрет/сек).	Аналогично.
W=<Выражение>	Установить задание контура скорости (об/мин).	Автоматически размыкается контур позиции. Используются текущие настройки контура тока и скорости.
W= <Выражение> A=<Выражение>	Установить задание контура скорости (об/мин), установить текущее ускорение.	Аналогично.
PCUR=<Выражение>	Установить текущую позицию в заданное значение.	Только при замкнутом контуре позиции.

Управление портами вывода.

Код команды	Описания	Примечания
PORT.x=<Выражение>	Установить текущее значение порта x в состояние <Выражение>.	Где x принимает значения от 0 до 3, <Выражение> принимает значение 0 или 1.

Вспомогательные функции

Код команды	Описания	Примечания
ABS(<Выражение>)	Функция возвращает абсолютное значение выражения	
D=<Выражение>	Ожидание истечения заданного интервала времени.	Время ожидания задается в миллисекундах. Максимальное – 32767 мс

Функция ожидания.

Код команды	Описания	Примечания
WAIT(<Условие>)	Ожидание выполнения условия <Условие>.	<Условие> - см. п. «Принятые обозначения»

Условные операции.

Код команды	Описания	Примечания
IF (<Условие>) <Операции> ENDIF	Выполнить набор операций <Операции>, если <Условие> имеет истинное значение.	<Условие> - см. п. «Принятые обозначения»

Циклические операции

Код команды	Описания	Примечания
WHILE (<Условие>) <Операции> ENDWHILE	Выполнять набор операций <Операции>, до тех пор, пока <Условие> имеет истинное значение.	<Условие> - см. п. «Принятые обозначения»

Допускается использование вложенных условных и циклических операций.

Примеры 1:

```
IF(P>4000)
    PORT.0=1
    D=1000
    PORT.1=0
    IF(P>8000)
        P=0
    ENDIF
ENDIF
```

Примеры 2:

```
WHILE(PORT.1 = 0x1)
    WHILE(P > 0x0FA0)
        X=X+1
    ENDWHILE
ENDWHILE
```

Функции передачи и приема по шине CAN.

Код команды	Описание	Примечание
SEND(ADDR, COM, VAR)	Передача команды COM по шине CAN устройству с адресом ADDR. В качестве 32-х битного аргумента команды передается значение одной из переменных X, Y или Z.	Передача сообщения выполняется асинхронно, т.е. после инициализации процесса передачи интерпретатор продолжает выполнение команды не дожидаясь завершения передачи команды.
SEND(ADDR, COM, CONST)	Передача команды COM по шине CAN устройству с адресом ADDR. В качестве аргумента команды передается константа CONST.	Аналогично
GET(ADDR, COM, VAR)	Получение команды COM по шине CAN от устройства с адресом ADDR. При этом 32-х битный аргумент записывается в одну из переменных X, Y или Z.	Прием сообщения выполняется синхронно, т.е. интерпретатор выходит из функции GET только после завершения приема команды с заданными параметрами.

Поле ADDR задается в диапазоне от 0 до 7. При этом адрес 7 является широко-вещательным, т.е. принимается всеми активными устройствами в сети.

Поле COM принимает значение от 21 до 31. Команды с кодами 0-20 используются блоком управления СПШ.

Завершение программы.

Код команды	Описания	Примечания
REPEAT	Повторить программу с нуля.	
HALT	Выйти из программы.	
PROGRAM N	Перейти на выполнение программы, находящейся в банке N.	Где N изменяется от 0 до 7.

Команда «PROGRAM» позволяет выполнять следующие типовые задачи:

- переход на основную программу движения после выполнения алгоритма поиска нулевой позиции;
- переход на выполнение программы по коду, выставленному на порте ввода, а также переход на подпрограммы экстренного останова и др.

События

Язык программирования СПШ10 предусматривает возможность асинхронного выполнения последовательности операций при наступлении определенных событий.

Операции, связанные с событиями

Код команды	Описание	Примечание
EVENT <Номер> <Условие>	Регистрация события в программе. По умолчанию не активно.	<Номер> - от 0 до 7
EVENT <Номер>+<Условие>	Регистрация события в программе. По умолчанию активно.	<Номер> - от 0 до 7
ON_EVENT <Номер> <Операции>	Программа обработки события. После обработки событие автоматически деактивизируется. Активация может выполняться явным образом с помощью команды +EVENT	<Номер> - от 0 до 7
+EVENT <Номер>	Разрешить реакцию на событие	<Номер> - от 0 до 7
-EVENT <Номер>	Запретить реакцию на событие	<Номер> - от 0 до 7

Для событий введен ряд дополнительных условных операторов. Список приведен в таблице ниже.

Дополнительные условные операции для событий

Код команды	Описание	Примечание
CAN_RECV (<Адрес>, <Команда>, VAR)	Асинхронный прием сообщения по шине CAN	
START	Начало выполнения программы	Событие возникает непосредственно перед выполнением первой команды программы

Пример использования событий

Строки текста программы

EVENT 0+START

EVENT 1 + PORT.1=1

EVENT 2 + CAN_RECV(1,22,X)

REPEAT

ON_EVENT 0

SEND(1,21,X)

ON_EVENT 1

-EVENT 1

Расшифровка/значение

Событие запуск программы

Событие установки порта в значение логической единицы

Событие приема данных по шине CAN

Основное тело программы соотит только из бесконечного цикла

Обработчик события 0

Обработчик события 1

Строки текста программы

Расшифровка/значение

P=P,A=1000
SEND(1,22,P)
ON_EVENT 2
P=X
+EVENT 1

Обработчик события 2

Встроенные переменные и функции

Встроенные переменные

Код команды	Описание	Примечание
\$drv_status	Параметр позволяет проанализировать и установить статус привода (см. параметр dd11 в документе «Описание параметров СПШ»).	
\$analog_inputA \$analog_inputB	Содержит текущее значение аналоговых портов ввода 0 и 1 соответственно.	Параметр только для чтения
\$eeprom_mas	Текущий элемент массива общего назначения, который расположен в энергозависимой памяти привода. Массив состоит из 1500 элементов, хранящие 32-х битные знаковые числа. Массив доступен как на чтение, так и на запись. При обращении к текущему элементу массива, происходит автоматическое смещение указателя на один элемент вперед. При обращении к элементу с номером 1499 индекс массива обнуляется.	
\$eeprom_mas_i	Установить текущий индекс массива энергозависимой памяти привода.	
\$ep3	Номинальный ток (см. документ «II-описание параметров.pdf»).	
\$ep4	Максимальный ток (см. документ «II-описание параметров.pdf»).	
\$ep7	Модуляция ШИМ сигнала (см. документ «II-описание параметров.pdf»).	

Встроенные функции

Код команды	Описание	Примечание
#rst_pos_i(dir)	Функция выполняет алгоритм поиска нулевой метки энкодера и обнуляет текущую позицию. Параметр, dir определяет направление вращения: 0 - по часовой стрелке, 1 – против часовой стрелки.	Функция выполняется корректно только если параметр st5 установлен в 1:Да.
#p_limit_l(pos) #p_limit_r(pos)	Установить левое/правое программное ограничение позиции. После установки программных ограничителей позиции привод предотвращает выход текущей позиции за их пределы.	
#crash_port(num, level)	Установить номер порта, который используется для аварийного останова привода. Период опроса состояния указанного порта составляет 25 мсек.	num - номер порта, принимает значение от 0 до 3. level - уровень, при котором происходит аварийное выключение привода, принимает значение 0 или 1.

Примеры программ

Программа выполнения циклической операции.

Строки текста программы

Расшифровка / значение

WAIT(PORT.0=0)

Ожидание внешнего сигнала управления

P=8000,W=100,A=10

Перемещение с указанными скоростями, ускорениями

PORT.0=1

Перевод порта вывода в активное состояние - запуск внешнего оборудования

WAIT(P>7995)

Ожидание достижения позиции

D=10

Задержка 10 мсек - ожидание завершения выполнения операций внешнего оборудования

P=0,W=20,A=10

Возврат к исходному положению на малой скорости

WAIT(P<4000)

Ожидание достижения промежуточной позиции

P=0,W=100,A=5

Изменение режима движения

WAIT(P<3)

Ожидание в заданной позиции

REPEAT

Повтор цикла с начала

Ниже приведена осциллограмма выполнения данной программы:

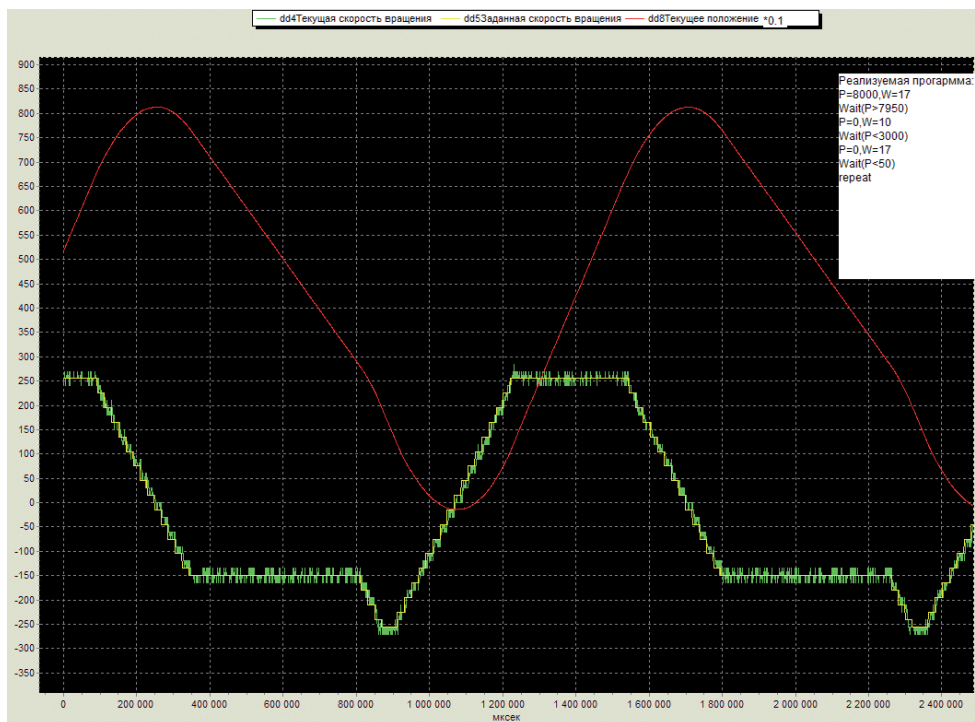


Рис. 1. Осциллограмма обработки программы иклических операций.

Программа задания постоянного режима вращения с ожиданием сигнала останова.

IF (PORT.1=0)

W=150, A=10

ENDIF

IF (PORT.1=1)

W=0, A=100

ENDIF

REPEAT

Программа поиска нулевой позиции по усилию.

W=150 Установить малую скорость вращения

WAIT(I>1000) Ожидание достижение тока заданного значения

PCUR=0 Обнуление текущей позиции

W=0 Установка скорости 0

P=0,W=300..... Установить позицию в 0

HALT Завершить программу

Программа поиска нулевой позиции по концевому датчику.

W=150 Установить малую скорость вращения

WAIT(PORT.0=0)..... Ожидание сигнала концевика

W=-10..... Установить ползучую скорость вращения
в обратном направлении

WAIT(PORT.0=1) Ожидание инверсного сигнала концевика

PCUR=0 Обнуление текущей позиции

W=0 Установка скорости 0

P=0,W=300..... Установить позицию в 0

PROGRAM 0..... Переход на программу циклического движения

Пример программы управления статусом.

В данной программе цифровой порт ввода 1 используется для аварийного останова и выхода из аварийного состояния, порт вывода 0 используется для индикации состояния привода.

IF(PORT 1=1) Если порт 1 активен - значит привод необходимо экстренно перевести в безопасное состояние.

A=10000 Устанавливаем максимальное ускорение.

P=PCUR Выставляет текущую позицию в качестве заданной.

WAIT(W=0)..... Ожидание завершения останова.

\$drv_status=1 Отключение генерации ШИМ.

X=1..... Запоминаем событие отключения.

ENDIF

IF(PORT 1=0)..... Порт не активен.

IF(X=1) Было ли отключение ШИМ?

X=0 Сброс события отключения.

D=1500..... Во избежание дребезга контактов на порте ввода внесем задержку 1,5 сек после включения привода.

\$drv_status=0..... Восстанавливаем генерацию ШИМ.

```

ENDIF
ENDIF
IF($drv_status=0).....Привод работает?
    PORT.0=0.....Вывод состояния ОК.
ENDIF
IF($drv_status>0) .....Привод не работает?
    PORT.0=1 .....Вывод статуса ошибки.
ENDIF
REPEAT
Пример использования массива, расположенного во внутренней энергонезависимой памяти привода для прохождения последовательности позиций:
IF(X=0)
    $eeprom_mas_i=0 ..... Установить указатель массива на 0-ую ячейку
ENDIF
Y=$eeprom_mas ..... Считать значение из текущей ячейки массива (После
                        чтения указатель массива автоматически смещается на
                        следующий элемент)
P=Y ..... Установить считанное значение а качестве задания
           контура позиции
WAIT(P=Y) ..... Ожидание достижения заданной позиции
PORT.0=1 ..... Включение внешнего технологического оборудования
X=X+1 ..... Индекс текущей позиции
IF(X>19) ..... В данном примере технологический цикл составляет
                20 позиций
    X=0 ..... Переходим к первой позиции
ENDIF
WAIT(PORT.0=1) ..... Ожидание завершения выполнения операции внеш-
PORT.0=0 ..... Сброс сигнала управления внешнего
                технологического оборудования
REPEAT

```

Программа передачи данных по шине CAN.

P=0 Установить задание 1 по положению
PORT.0=0 Установить порт в 0.
WAIT(P<10) Ожидание достижения заданной позиции.
X=P Считывание текущей позиции.
SEND(0,21,X) Передача сообщения о достижении первой позиции.
P=8000 Установить задание 2 по положению.
PORT.0=1 Установить порт в 1.
WAIT(P>7980) Ожидание достижения заданной позиции.
X=P Считывание текущей позиции.
SEND(0,22,X) Передача сообщения о достижении второй позиции.
REPEAT

После разработки программы рекомендуется сохранять программу на диск компьютера с целью возможности восстановления программ после потенциальной потери данных энергонезависимой памяти. Такая ситуация возможна при выходе двигателя из строя, при импорте параметров привода.

Интегрированный сервопривод СПШ10

III - язык программирования



Сервотехника

«Сервотехника» ЗАО
Выборгская ул., д. 22
125130 Москва
Россия
Тел.: (495) 797-88-66
Факс: (495) 450-00-43
info@servotechnica.ru
www.servotechnica.ru