



## Содержание

ГЛАВА 1 Введение .....	3
1.1 Краткое описание .....	3
1.2 Список моделей .....	3
1.3 Условия эксплуатации .....	3
ГЛАВА 2 CPU .....	5
2.1 Обзор .....	5
2.1.1 Внешний вид .....	5
2.1.2 Типы CPU .....	5
2.2 Функции .....	6
2.2.1 Состояние CPU и LED .....	6
2.2.2 Порт программирования и последовательный порт .....	7
2.2.3 Ethernet порт .....	8
2.2.4 CAN порт .....	8
2.2.5 Модули расширения .....	8
2.2.6 Высокоскоростной счетчик и высокоскоростной импульсный выход .....	9
2.2.7 Прерывания по фронту .....	9
2.2.8 Сохранение данных и резервное копирование данных .....	9
2.2.9 Часы реального времени (RTC) .....	10
2.2.10 Батарея резервного питания .....	10
2.3 Схема подключения .....	11
2.4 Размеры .....	15
2.5 Технические характеристики .....	15
ГЛАВА 3 Модули расширения .....	17
3.1 Обзор .....	17
3.2 Цифровые .....	18
3.2.1 KS121-16DX .....	18
3.2.2 KS122-14DT .....	20
3.2.3 KS122-12XR .....	22
3.2.4 KS123-14DR .....	23
3.3 Аналоговые .....	25
3.3.1 KS131-04RD .....	25
3.3.2 KS133-06IV .....	27
ГЛАВА 4 Программное обеспечение .....	29
4.1 Протокол Modbus TCP для связи с сторонними устройствами .....	29
4.1.2 Регистры Modbus .....	29
4.2 Высокоскоростной счетчик .....	30
4.2.1 Режимы работы и входы высокоскоростных счетчиков .....	30
4.2.2 Байт управления и Байт состояния .....	31
4.2.3 Установка заданного значения (значение PV) .....	32
4.2.4 Номер события "CV = PV" .....	34
4.2.5 Как использовать высокоскоростной счетчик .....	34
4.3 Как использовать высокоскоростной импульсный выход .....	36

4.3.1 Инструкция высокоскоростного импульсного выхода .....	36
4.3.2 Как использовать инструкцию PLS .....	37
4.3.3 Как использовать команды управления позицией .....	42
4.4 Использование CAN шины .....	44
4.4.1 Подключение .....	44
4.4.2 Функция шины расширения .....	45
4.4.3 Функция Kinco Motion Control .....	46
4.4.4 Функция CANOpen Master .....	48
4.4.5 Функция CANOpen slave .....	55
4.4.6 Функция связи CAN Free .....	56
4.5 Команды CANBus .....	57
4.5.1 Команды по управлению движением Kinco .....	57
4.5.2 Команды SDO .....	73
4.5.3 Команды CAN_Free .....	76

## ГЛАВА 1 Введение

### 1.1 Краткое описание

ПЛК Kinco серии KS является небольшим и интегрированным контроллером. Это новый, тонкий и высокопроизводительный ПЛК Kinco. Основан на высокой производительности, высокой надежности и мощных функциях K5 / K2, серия KS использует процессор более высокого уровня. ПЛК KS имеет порт CANopen, более высокоскоростной ввод и вывод, малый размер для установки.

### 1.2 Список моделей

Серия	Модель	Описание
CPU101	KS101M-04DX	Напряжение питания DC24V, DI 4*DC24V, 1*MicroUSB, 1*Ethernet 1*RS232 порт для программирования, 1*RS485, 2*CAN Модули расширения: max 14
CPU105	KS105-16DT	Напряжение питания DC24V, DI 8*DC24V, DO 8*DC24V 1*RS232 порт для программирования, 1*RS485 Модули расширения: max 14
	KS105C1-16DT	Напряжение питания DC24V, DI 8*DC24V, DO 8*DC24V 1*RS232 порт для программирования, 1*RS485, 1*CAN Без поддержки модулей расширения.
	KS105C2-16DT	Напряжение питания DC24V, DI 8*DC24V, DO 8*DC24V 1*RS232 порт для программирования, 1*RS485, 2*CAN Модули расширения: max 14

### 1.3 Условия эксплуатации

Kinco-KS соответствует стандарту GB/T 15969.3-2007 (idt IEC61131-2: 2007) и характеристикам испытаний. В следующей таблице перечислены все условия и требования для правильной работы. Пользователь обязан убедиться, что условия эксплуатации не выходят за границы.

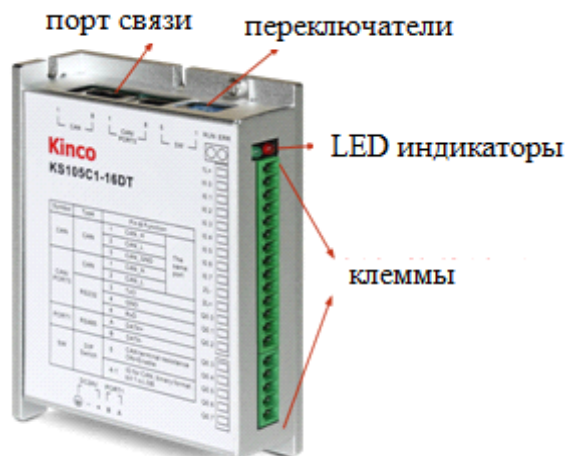
Транспортировка и хранение		
Условия окружающей среды	Температура	-40°C ~ +70°C
	Относительная влажность	10% ~ 95%, без конденсата
	Высота	До 3000м
Нормальная работа		
Условия окружающей среды	Температура воздуха	Открытое оборудование: -10°C ~ + 55°C Закрытый оборудование: -10°C ~ + 40°C
	Относительная влажность	10% ~ 95%, без конденсата
	Высота	До 2000м
Механические условия	Синусоидальная вибрация	5 < F < 8,4, Случайные: амплитуда 3,5 мм; Непрерывная: амплитуда 1.75mm. 8.4 < F < 150, Случайные: 1,0g ускорение; Непрерывная: 0,5g ускорение.

Электромагнитная совместимость (EMC)	Электростатический разряд	± 4 кВ контакт, ± 8 кВ воздух. Исполнение В.
	Скачок напряжения	АС питание сети: 2KV CM, 1KV DM; DC питание сети: 0.5KV CM, 0.5KV DM; I/O и порт связи: 1KV CM. Исполнение В.
	Быстрые переходные импульсы	Питание сети: 2KV, 5KHz. I/O и порт связи: 1KV, 5KHz. Исполнение В.
	Падение и пропадание напряжение питания	Переменный ток питания: 50 Гц, 0% напряжения 1 период; 40% напряжения на 10 периодов; 75% напряжения на 20 периодов. Исполнение А.
Степень защиты		IP20

## Глава 2 CPU

### 2.1 Обзор

#### 2.1.1 Внешний вид



#### 2.1.2 Типы CPU

Kinco-KS имеет различные модели CPU с разнообразием функций и возможностей, все CPU используют источник питания DC24V. В следующей таблице описаны основные технические данные каждой модели CPU.

Параметр	KS101M-04DX	KS105-16DT	KS105C1-16DT	KS105C2-16DT
Источник питания				
Номинальное напряжение питания	24VDC.			
Диапазон напряжения питания	20.4VDC ~ 28.8VDC			
I/O				
Цифровые I/O	4*DI	8*DI, 8*DIO		
Аналоговые I/O	---	---	---	---
Модуль расширения	14	14	---	14
Ethernet	1	---	---	---
CAN	2	---	1	2
Порт связи	PORT0, RS232, протокол программирования, MODBUS RTU slave, free протокол PORT1, RS485, протокол программирования, MODBUS RTU master, free протокол			
Высокоскоростные счетчики	2, макс. 200 кГц	4, макс. 200 кГц, поддержка однофазного и двухфазного сигнала		
Высокоскоростной импульсный выход	---	4 Канал 0, 1 и 2 Макс. 200 кГц (сопротивление нагрузки составляет менее 1,5 KΩ на самой высокой частоте) Канал 3 Макс. 10 кГц		

Прерывания	---	4, I0.0 ~ I0.3.
<b>Область памяти</b>		
Максимальная программа пользователя	8К инструкций	4К инструкций
Данные пользователя	M область: 4 Kbyte; V область: 16 Kbyte	M область: 1 Kbyte; V область: 4 Kbyte
Резервное копирование данных	E2PROM, последний 1 Кбайт области V	E2PROM, 448 bytes
Диапазон сохранения	V область, литиевый элемент, 3 года при нормальной температуре.	1 Кбайт, литиевый элемент, как резервный источник питания, 3 года при нормальной температуре.
<b>Другое</b>		
Таймеры	256 1 мс времени база: 4 10 мс времени база: 16 100 мс времени база: 236	
Временные прерывания	2 с базой 0,1 мс	
Счетчики	256	
Часы реального времени	Да, отклонение менее 5 мин / месяц при 25°C	

## 2.2 Функции

### 2.2.1 Состояние CPU и LED

CPU имеет два режима: режим **STOP** и режим **RUN**. В режиме **RUN**, процессор выполняет основной цикл сканирования и все задачи прерывания. В режиме **STOP**, процессор будет устанавливать все выходы (в том числе DO и AO) в значения, указанные в [**Hardware Configuration**] через Kincobuilder, и только запросы процесса связи, которые исходят от программного обеспечения KincoBuilder и других Modbus RTU ведущего устройства.

— Изменение состояния CPU

Kinco-KS имеет два способа для ручного изменения состояния CPU:

- Установите все переключатели CAN-порта в положение [OFF], ПЛК будет находиться в состоянии STOP. Если любой переключатель [ON], ПЛК будет в состоянии RUN. (для KS105 используйте все 5 переключателей, для KS105C1 используйте 1 ~ 4 переключатели, для KS105C2 используйте 1 ~ 3).
- Используйте переключатель (RUN / STOP) в Kincobuilder; Выполните команды [Debug] -> [RUN] или [STOP].

В следующей таблице перечислены результаты этих двух способов.

Переключатель	Команда KincoBuilder	Фактический режим работы
RUN	RUN	RUN
	STOP	STOP
STOP	RUN	STOP
	STOP	STOP

Кроме того, если CPU обнаружит любую серьёзную ошибку, он немедленно перейдёт в состояние **STOP**.

— Индикация состояния CPU

Модуль CPU имеет четыре состояния LED индикаторов: **RUN**, **STOP**, **Comm.** и **Err.**

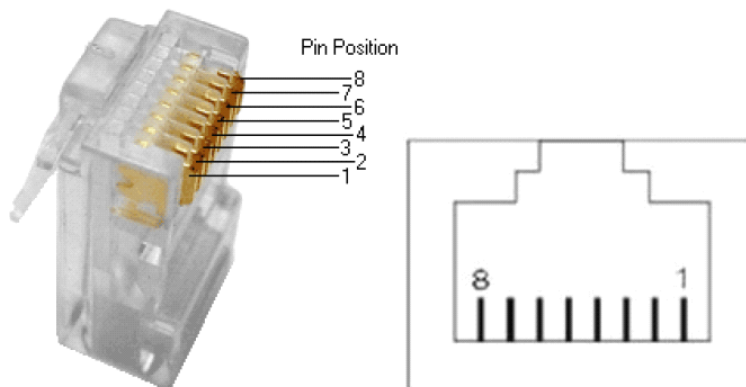
**[RUN]** : Если CPU находится в состоянии **RUN**, зелёный LED индикатор будет включен. Если CPU находится в состоянии **STOP**, он будет выключен.

**[Err.]**: Если CPU обнаружит ошибку в программе пользователя или модуля расширения, включится красный LED индикатор.

KS разделяет ошибки по трём уровням: фатальная ошибка, серьезная ошибка, нормальная ошибка. Когда CPU обнаружит ошибку, он будет использовать другой способ управления в соответствии с уровнем ошибки и включит индикатор **Err.**, потом он сохранит код ошибки для анализа пользователя.

### 2.2.2 Порт программирования и последовательный порт

KS обеспечивает 2 порта связи, PORT1 и PORT2. Они поддерживают скорость до 115,2 кбит / с. PORT1 может использоваться как порт программирования, а также поддерживать протокол Modbus RTU slave и протокол free. PORT2 поддерживает протокол Modbus RTU (как ведомый или ведущий) и протокол free. Порт программирования PS232 находится в порту RJ45. Выводы и функции, как показано ниже.



PC (RS232)	Контакт		Контакт	PLC (RJ45)
Rx	2	-----	3	Tx
Tx	3	-----	6	Rx
GND	5	-----	4	GND

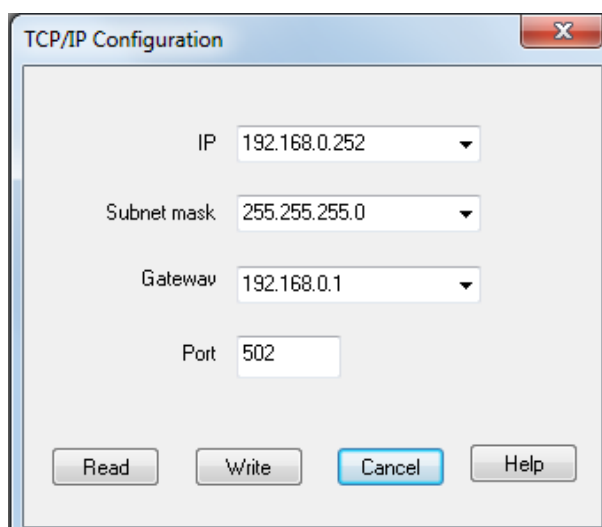
**!** RS232 (RJ45) не должен подключаться / отключаться под напряжением, поэтому необходимо отключить питание ПЛК или ПК, иначе порт будет повреждён.

### 2.2.3 Ethernet порт

ПЛК KS101M-04DX имеет интерфейс Ethernet, соответствующий стандарту IEEE802.3. Этот интерфейс поддерживает протокол программирования и может использоваться как порт программирования. Кроме того, он поддерживает сервер MODBUS TCP, широко известный как «slave» функция.

Кабель связи может быть прямым (straight wire) или перекрестным кабелем (cross wire). Интерфейс Ethernet обеспечивает функцию «автосогласования». Когда кабель вставлен, ПЛК автоматически определит тип используемого кабеля.

Пользователи могут изменять параметры интерфейса Ethernet через порт USB, последовательный порт или сам порт Ethernet: выполните команду меню **[Tool] - [TCP / IP parameter configuration]** в программном обеспечении KincoBuilder, появится следующее диалоговое окно, и пользователь может читать или изменять параметры.



В локальной сети ПЛК и ПК, расположенные в одном сегменте сети (т.е. первые три числа IP-адреса должны быть одинаковыми, а последний - другим), могут обмениваться данными друг с другом.

### 2.2.4 CAN порт

KS105C1-16DT имеет 1 порт CAN. Он может поддерживать команды "motion control", CANopen master / slave и free протокол.

KS105C2-16DT / KS101M-04DX имеют 2 порта CAN, CAN1 и CAN2. CAN2 может поддерживать команды "motion control", CANopen master и free протокол. CAN1 может поддерживать протокол free или модули расширения.

### 2.2.5 Модули расширения

KS105-16DT имеет порт расширения, он может подключать модули расширения серии KS. Порт CAN1 KS105C2-16DT / KS101M-04DX может работать как порт расширения, или как CAN порт с поддержкой свободного протокола. Пользователи могут использовать их напрямую без настройки, ПЛК может идентифицировать их автоматически. Модули расширения имеют один входной порт и один выходной порт для удобства подключения нескольких модулей расширения. Пожалуйста, обратитесь к рисунку ниже. Для подключения модулей расширения нет необходимости указывать адрес. Реальный адрес в сети зависит от очереди в KincoBuilder в окне **Hardware**.





### 2.2.6 Высокоскоростной счетчик и высокоскоростной импульсный выход

KS имеет 4 высокоскоростных счетчика (HSC0 ~ HSC3). Высокоскоростной счетчик поддерживает несколько режимов: однофазный, CW / CCW, фаза AB. Все HSC поддерживают до 200 кГц (однофазные и фазы AB).

KS105 имеет 4 высокоскоростных импульсных выхода (Q0.0, Q0.1, Q0.4 и Q0.5). Все они поддерживают PTO и PWM. Q0.0, Q0.1 и Q0.4 поддерживает до 200 кГц (сопротивление нагрузки должно быть меньше, чем 1,5 кОм), Q0.5 поддерживает до 10 кГц.

### 2.2.7 Прерывания по фронту

I0.0 ~ I0.3 поддерживают функции прерывания по фронту, он может выполнить прерывание по переднему фронту и заднему фронту входного сигнала. С помощью этой функции, он может быстро захватить передний фронт и задний фронт входного сигнала. Для некоторых входных сигналов, ширина импульса меньше, чем время сканирования CPU, он может реагировать быстрее.

### 2.2.8 Сохранение данных и резервное копирование данных

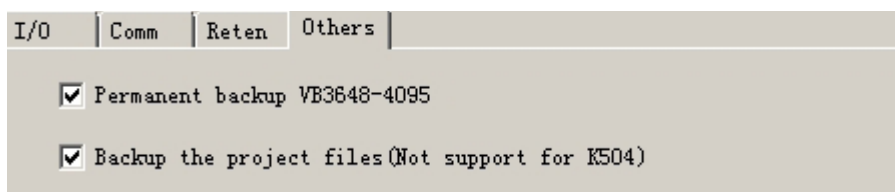
Сохранение данных означает, что данные в памяти могут сохраниться после сбоя питания. CPU имеет литиевую батарейку (заменяемая но не перезаряжаемая) для сохранения данных. Когда пропадает питание CPU, данные в оперативной памяти будут поддерживаться с помощью литиевой батарейки, и сохраняющиеся диапазоны будут оставлены без изменений до следующего включения. С помощью [Hardware] в конфигурации KincoBuilder, пользователь может выбрать тип сохраняемых данных (например, V, C) и диапазон области. Срок службы батареи составляет 5 лет, а продолжительность сохранения 3 года

при нормальной температуре.

Резервное копирование данных заключается в том, что CPU обеспечивает E2PROM постоянно сохранёнными данными. При включении питания, CPU будет восстанавливать данные из оперативной памяти E2PROM для исполнения.

**Примечание:** Поскольку E2PROM имеет ограничение на запись в 1 млн раз, пользователи должны избегать записи данных в области резервного копирования данных слишком часто.

Для резервного копирования данных в V области есть 448 байт (VB3648 ~ VB4095), данные в этой области будут сохраняться в E2PROM автоматически. KS устанавливает VB3648 ~ VB3902 как резервное копирование данных по умолчанию, если пользователь нуждается в использовании VB3903 ~ VB4095 для резервного копирования данных, то необходимо настроить в конфигурации **[PLC hardware configuration]**. Интерфейс конфигурации на следующем рисунке.



## 2.2.9 Часы реального времени (RTC)

Часы реального времени имеются во всех моделях CPU, могут отображать в режиме реального времени индикацию часов / календаря. Пользователи должны использовать KincoBuilder [PLC] → [Time of Day Clock...], чтобы установить часы при использовании RTC в первый раз. Затем пользователи могут использовать в режиме реального времени инструкции часов (READ\_RTC, SET\_RTC, RTC\_W, RTC\_R). После выключения питания CPU, часы реального времени могут поддерживаться литиевой батареей. Срок службы батареи составляет 5 лет, а продолжительность сохранения 3 года при нормальной температуре.

## 2.2.10 Батарея резервного питания

KS может использовать литиевую батарею в качестве резервного аккумулятора. Когда выключается питание PLC, он будет использовать резервную батарею для поддержания часов реального времени и RAM.

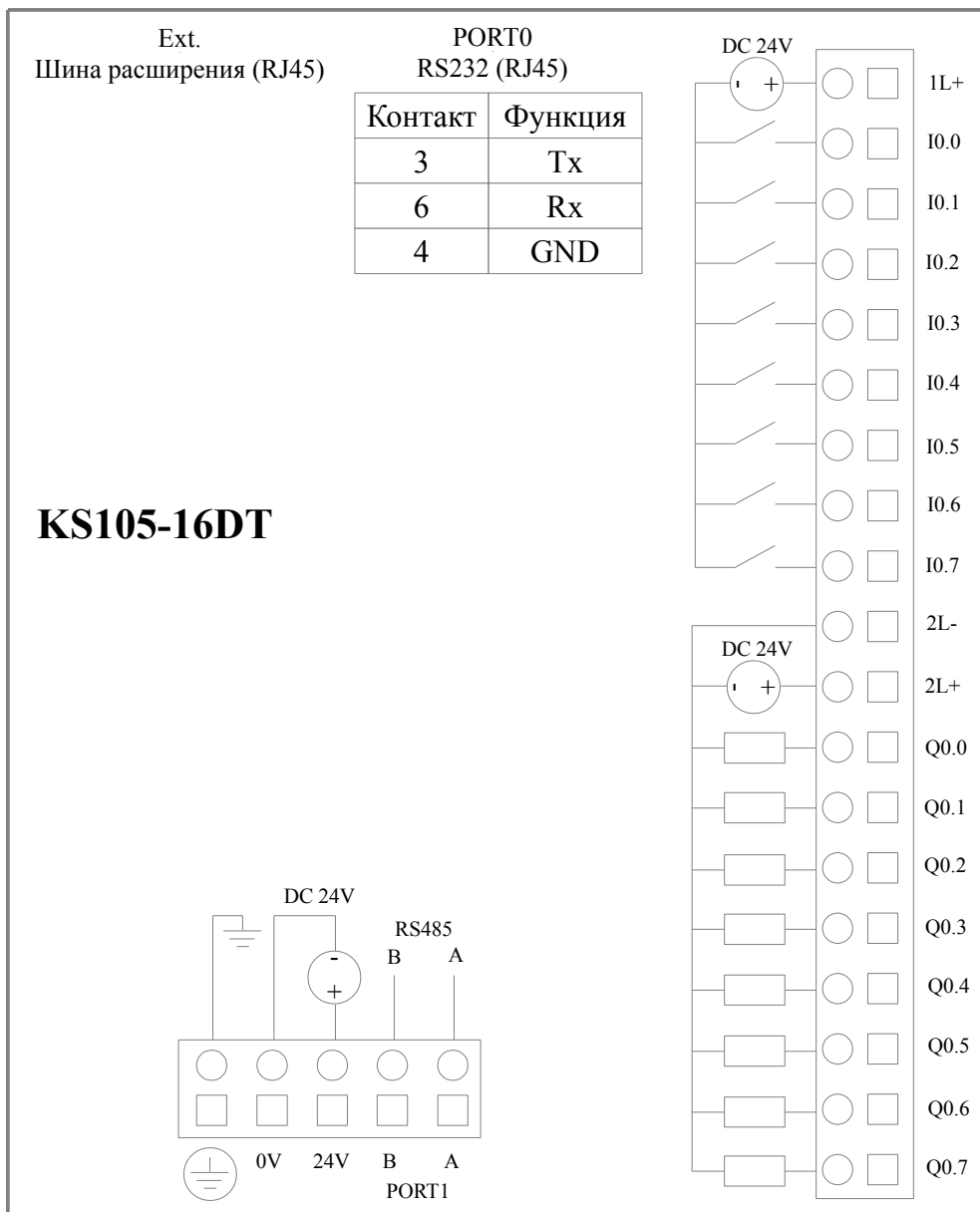
Резервная батарея съемная, пользователь может заменить батарею на новую, когда батарея разрядится.

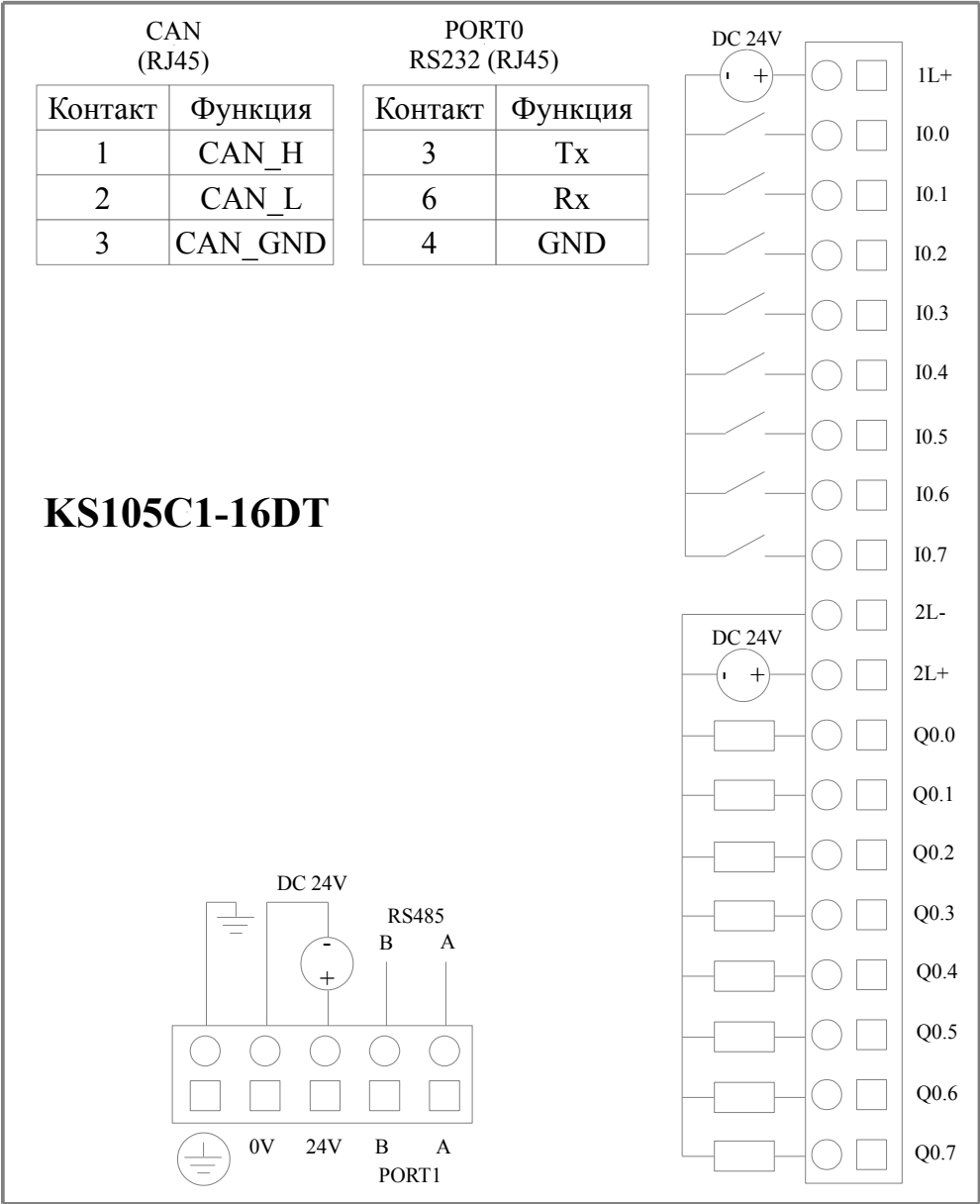
Аккумулятор литиевый CR2032 (3V) с разъемом, как показано на рисунке.

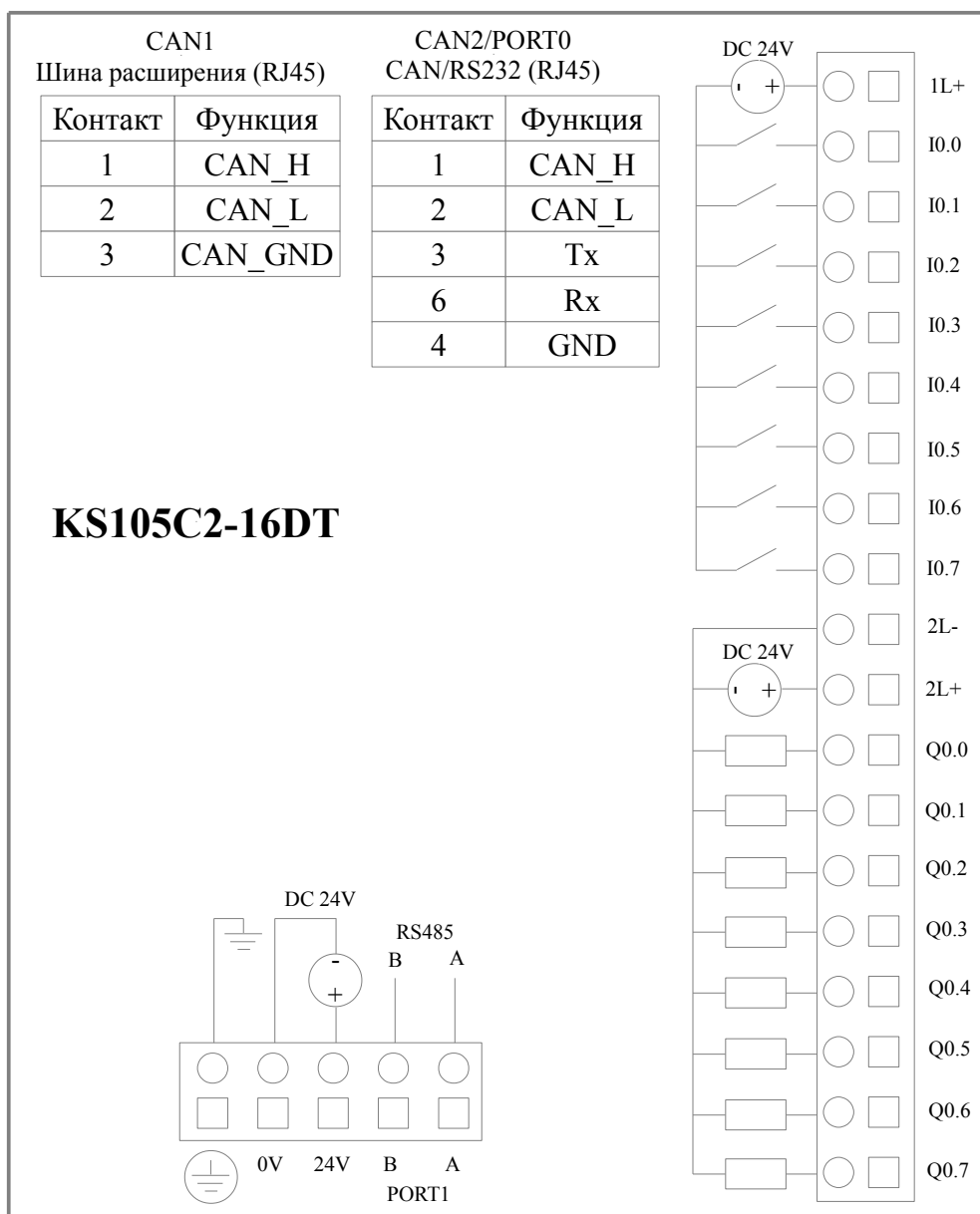
Пользователь может заказать батарею отдельно.

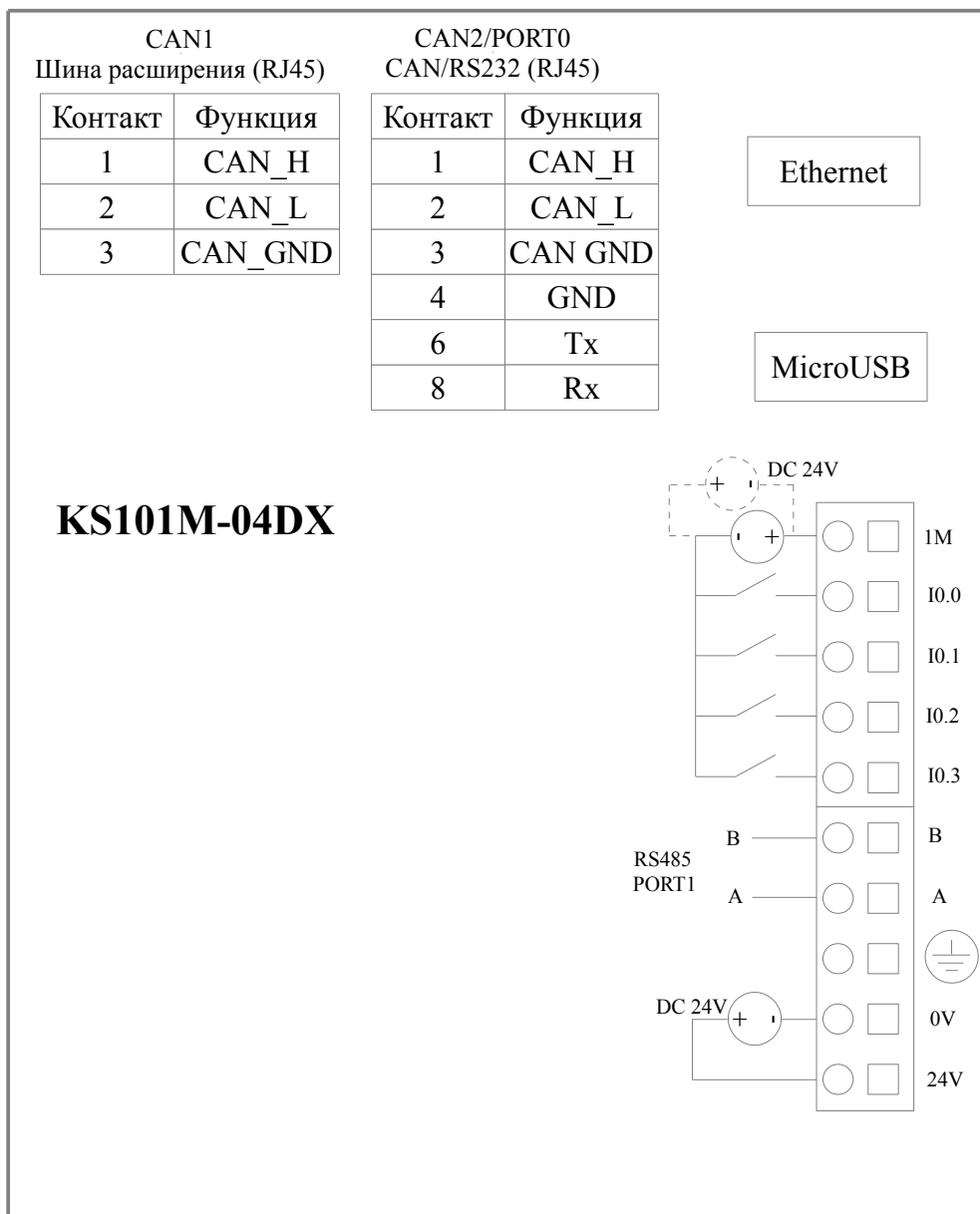


## 2.3 Схема подключения

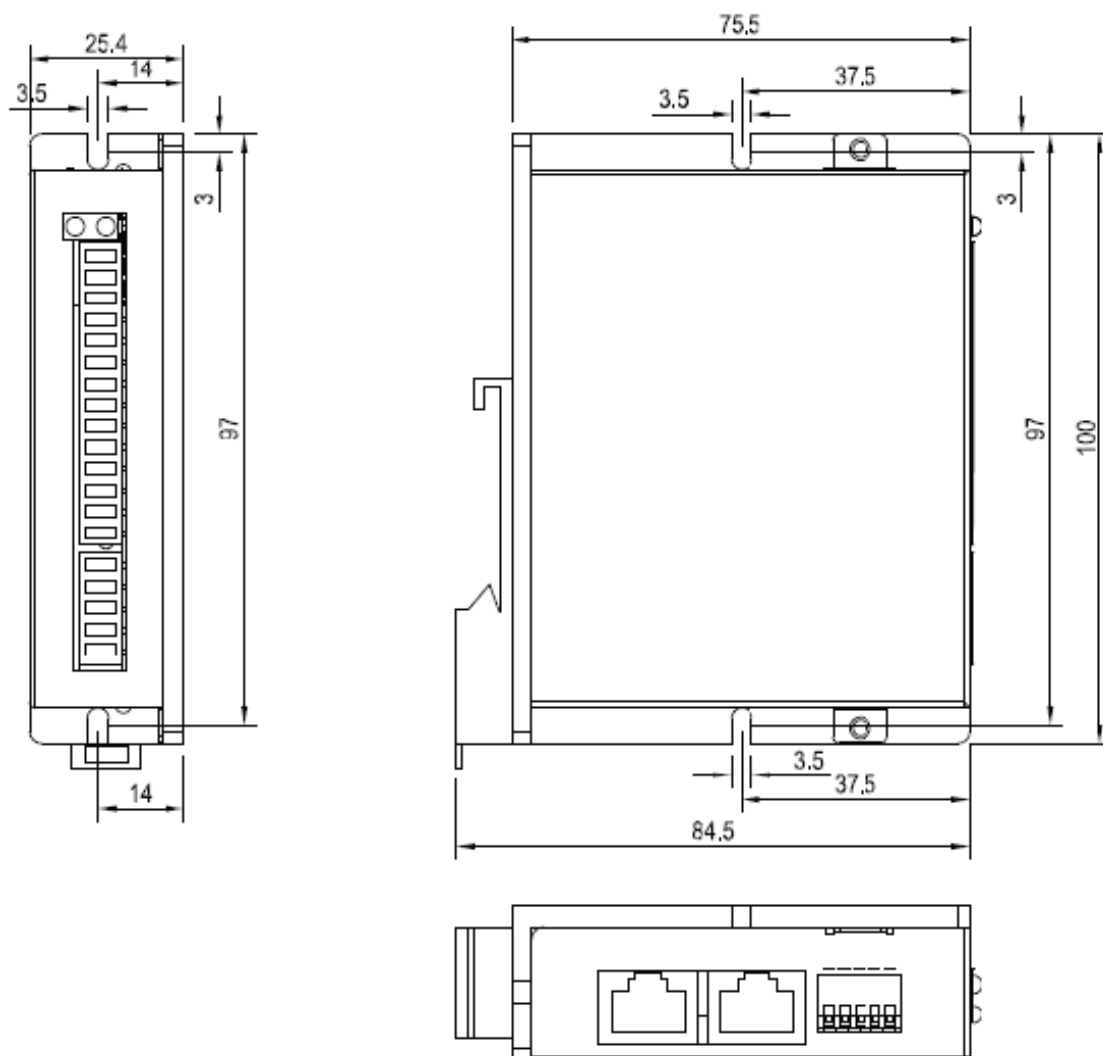








## 2.4 Размеры



## 2.5 Технические характеристики

### — Характеристики DI

Тип входа	Сток
Номинальное входное напряжение	DC 24V (Max. 30V)
Номинальный входной ток	3.5mA, 24VDC
Макс входное напряжение логического 0	5V, 0.7mA
Мин входное напряжение логической 1	Общий канал: 11V@2.0mA
Входной фильтр задержки времени	
· off-to-on	1,2 мкс
· on-to-off	0,5 мкс
Изоляция между входом и внутренними цепями	
· Режим	Оптоэлектрическая развязка
· Напряжение	500B AC / 1 мин

→ **Характеристики DO (транзисторный тип)**

Тип выхода	Исток
Номинальное напряжение питания	24В, допустимый диапазон: DC20.4V ~ DC28.8V <b>Примечание:</b> Питание транзисторного выхода не имеет схемы фильтрации и защитной схемы. Если напряжение превышает допустимый диапазон, это может привести к повреждению внутренних компонентов. Если источник питания нестабилен, убедитесь, что напряжение между 1L + и 1L- находится в допустимом диапазоне, при этом среднее напряжение не должно превышать 28,8 В, максимальное пиковое напряжение не должно превышать 32 В.
Выходной ток для каждого канала	Номинальный ток: 200 мА Max. ток 300 мА, 24VDC
Мгновенный импульсный ток для каждого канала	1А, менее 1 сек
Ток утечки выхода	Max.0.5А
Выходное сопротивление	Max. 0.2Ω
Задержка выхода · off-to-on · on-to-off	Общий канал: 12мкс; HSC канал: 0,5 мкс; Общий канал: 35мкс; HSC канал: 1мкс;
Защита: · Защита от неправильной полярности источника питания · Защита при индуктивной нагрузке · Защита от короткого замыкания · Защита от неправильной полярности выхода	Нет  Да Да Да, меньше, чем 10 сек.
Изоляция между выходом и внутренними цепями · Режим · Напряжение	Оптоэлектрическая развязка 500В AC/ 1 мин



## ГЛАВА 3 Модули расширения

### 3.1 Обзор

Серия KS может расширяться с помощью модулей расширения. Питание для модулей расширения KS составляет 24 В постоянного тока. Модули расширения KS имеют один стандартный порт RS485, который может работать как Modbus slave.

**Примечание:** Некоторые адреса ведущего устройства Modbus RTU начинаются с 1, поэтому данные в приведенной ниже форме должны добавлять 1 напрямую.

Тип	Форма	Код Modbus	Область	Modbus адрес
KS121-16DX	DI (1XXXX)	02	I0.0 --- I1.7	0-15
KS122-14DT	DO (0XXXX)	01, 05, 15	Q0.0-Q1.5	0-13
KS122-12XR			Q0.0-Q1.3	0-11
KS123-14DR	DI (1XXXX)	02	I0.0 --- I0.7	0-7
	DO (0XXXX)	01, 05, 15	Q0.0-Q0.5	0-5
KS131-04RD	AI (3XXXX)	04	AIW0 --- AIW6	0-3
KS133-06IV	AI (3XXXX)	04	AIW0 --- AIW6	0-3
	AO (4XXXX)	03, 06, 16	AQW0 --- AQWQ	0-1

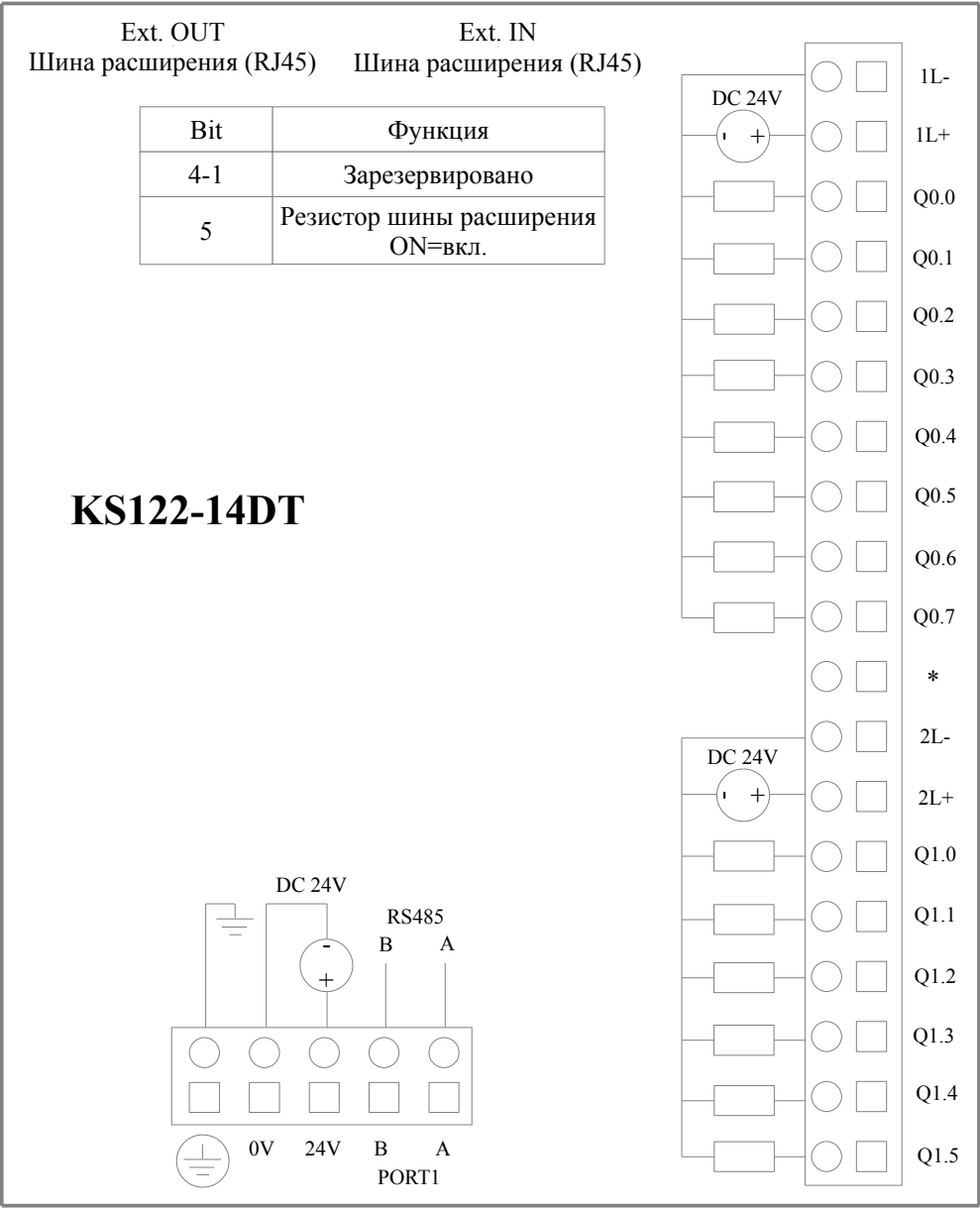


## Технические характеристики KS121-16DX

Электрические параметры	
Номинальное напряжение питания	DC 24V , $\geq 100\text{mA}$
Входы	16
Тип входа	исток / сток
Номинальное входное напряжение	DC 24V
Номинальный входной ток	3.5mA@24VDC
Максимальное входное напряжение логического "0"	5V
Минимальное входное напряжение логической "1"	11V@2.0mA
Время задержки входа · Задержка подключения · Задержка отключения	12 $\mu\text{s}$ 40 $\mu\text{s}$
Изоляция между входом и внутренними цепями - Тип изоляции - Напряжение изоляции	Фотоэлектрическая пара 500VAC/1мин
Размеры и вес	
Размеры (длина × ширина × высота)	100×84.5×25.4 мм
Вес	200г.

3.2.2 KS122-14DT

Этот модуль расширения имеет 14 транзисторных выходов.

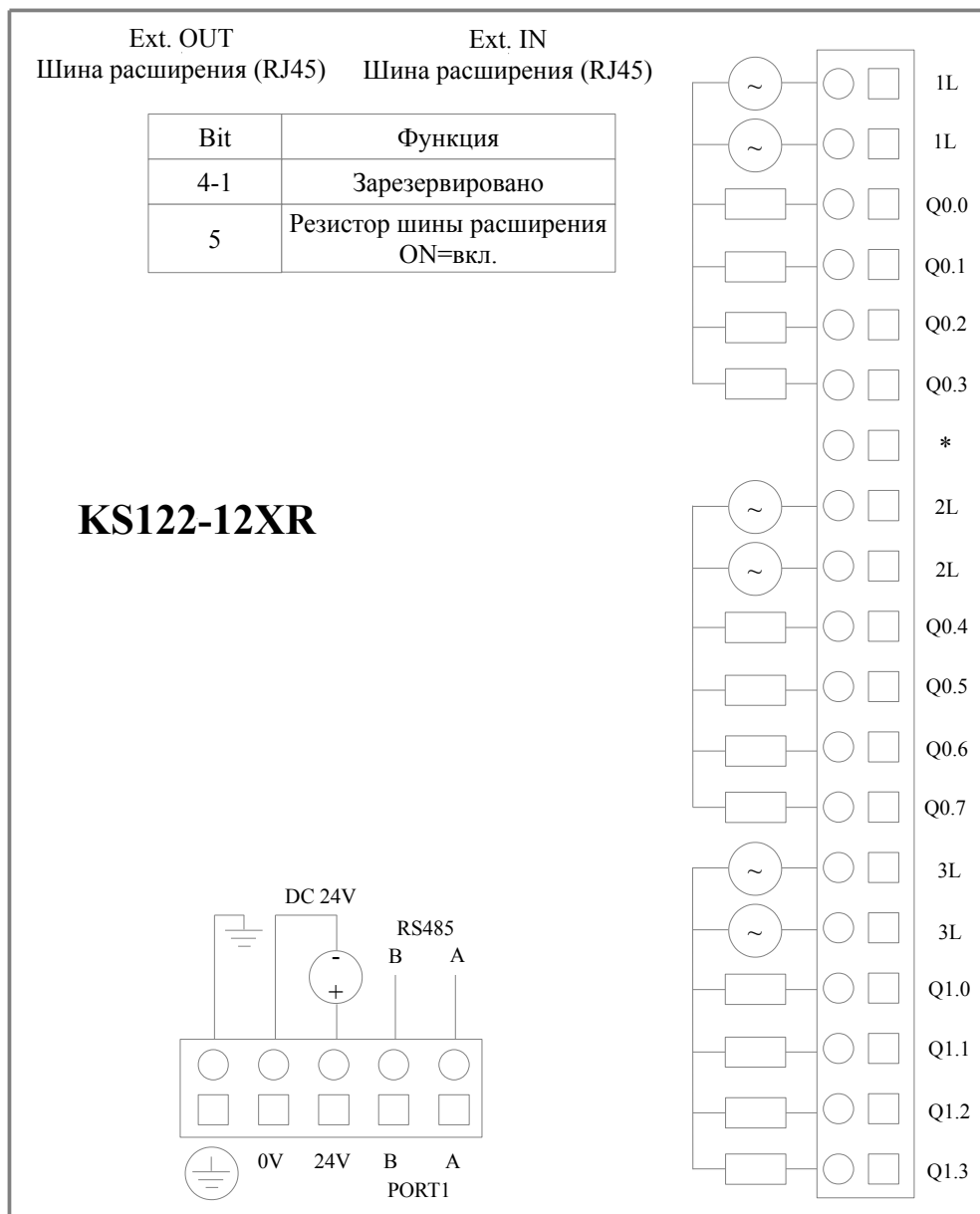


## Технические характеристики KS122-14DT

Электрические параметры	
Номинальное напряжение питания	DC 24V , $\geq 100\text{mA}$
Выходы	8
Тип выхода	исток
Номинальное выходное напряжение	DC24V. Допустимый диапазон: DC20,4V — DC28,8V
Выходной ток на канал	Max 500mA@24VDC
Выходной ток утечки	Max 0.5mA
Выходное сопротивление	Max 0.2Ω
Время задержки выхода · Задержка включения · Задержка отключения	0.3--5μs 5μs
Защитная функция: · Защита от полярности питания · Защита выхода индуктивной нагрузки · Защита от короткого замыкания · Защита от обратной полярности выхода	Да Да Да Да, не более 10сек.
Изоляция между выходом и внутренней схемой · Метод изоляции · Напряжение изоляции	Оптрон 500 В переменного тока / 1 минута
Размеры и вес	
Размеры (длина × ширина × высота)	100×84.5×25.4 мм
Вес	200г.

**3.2.3 KS122-12XR**

Этот модуль расширения имеет 12 релейных выходов.

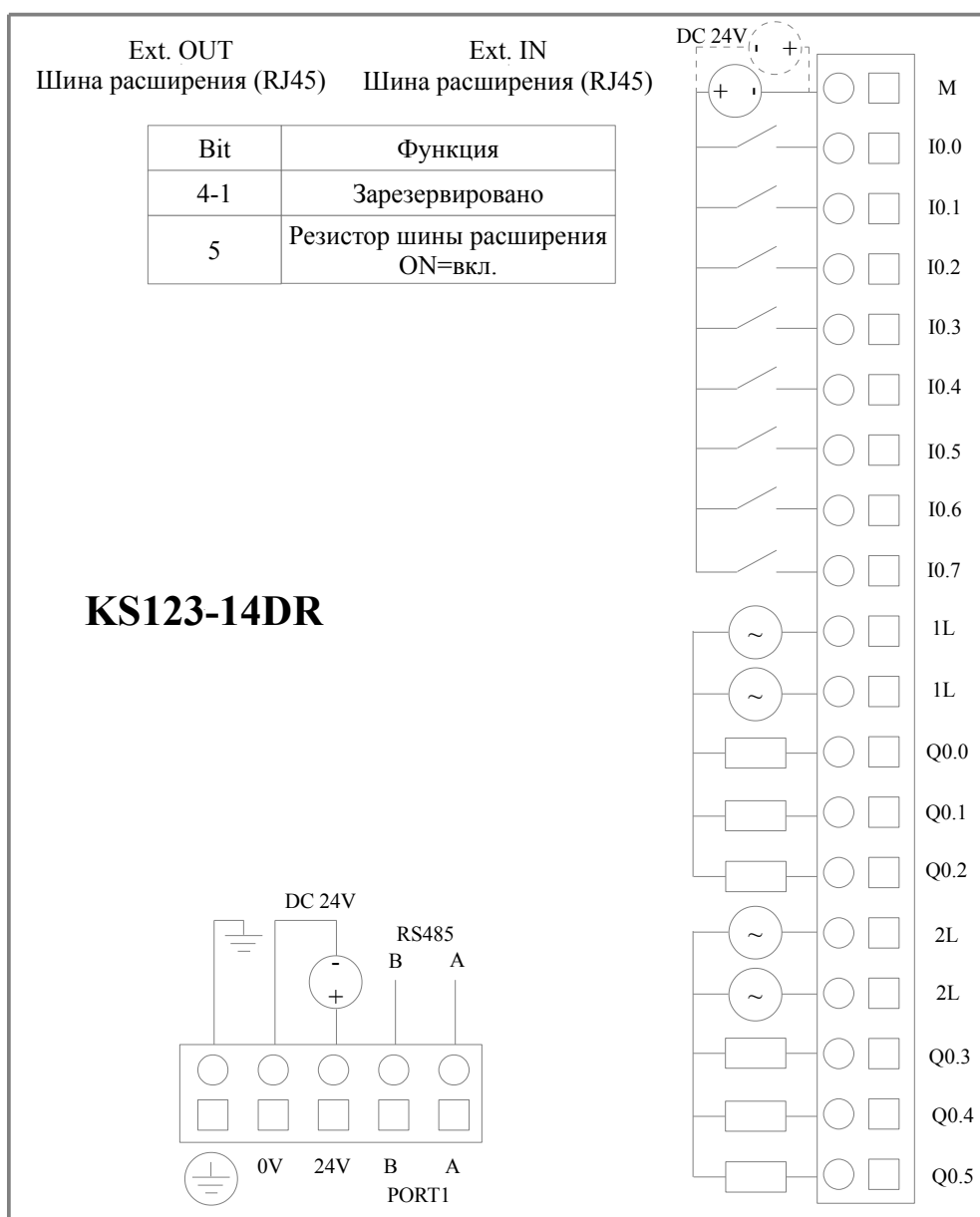
**Технические характеристики KS122-12XR**

Электрические параметры	
Номинальное напряжение питания	DC 24V , $\geq 100\text{mA}$
Выходы	12 реле
Максимальное напряжение нагрузки	DC 30V/AC250V
Выходной ток	2A ( DC 30V/AC250V )
Время задержки для включения выхода	10ms ( max. )
Время задержки для выключения выхода	5ms ( max. )
Срок службы реле - Механическая жизнь (без нагрузки)	20 , 000 , 000 раз ( 1200 раз/мин )

- Электрический ресурс (номинальная нагрузка)	100 , 000 раз ( 6 раз/мин )
Изоляция вывода · Режим изоляции · Напряжение изоляции	реле 2000Vrms
<b>Размеры и вес</b>	
Размеры (длина × ширина × высота)	100×84.5×25.4 мм
Вес	200г.

### 3.2.4 KS123-14DR

Этот модуль расширения имеет 8 цифровых входов и 6 релейных выходов.



## Технические характеристики KS123-14DR

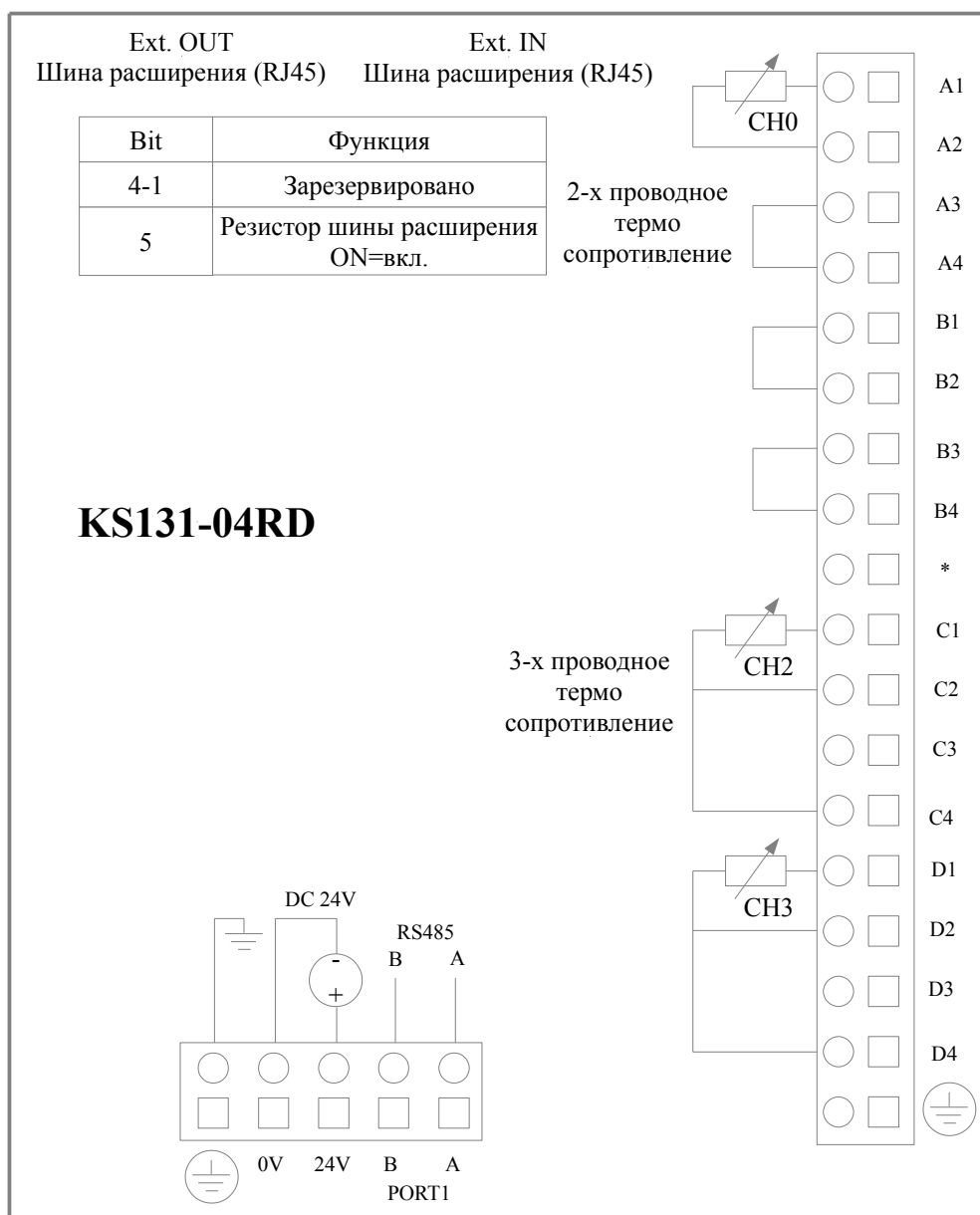
Электрические параметры	
Номинальное напряжение питания	DC 24V , $\geq 100\text{mA}$
Входы	8
Тип входа	исток / сток
Номинальный входной ток	3.5mA@24VDC
Максимальное входное напряжение логического "0"	5V
Минимальное входное напряжение логической "1"	11V@2.0mA
Изоляция между входом и внутренними цепями - Тип изоляции - Напряжение изоляции	Фотоэлектрическая пара 500VAC/1мин
Выходы	6 реле
Максимальное напряжение нагрузки	DC 30V/AC250V
Выходной ток	2A ( DC 30V/AC250V )
Время задержки для включения выхода	10ms ( max. )
Время задержки для выключения выхода	5ms ( max. )
Срок службы реле - Механическая жизнь (без нагрузки) - Электрический ресурс (номинальная нагрузка)	20 , 000 , 000 раз ( 1200 раз/мин ) 100 , 000 раз ( 6 раз/мин )
Изоляция вывода · Режим изоляции · Напряжение изоляции	реле 2000Vrms
Размеры и вес	
Размеры (длина × ширина × высота)	100×84.5×25.4 мм
Вес	200г.



### 3.3 Аналоговые

#### 3.3.1 KS131-04RD

Модуль имеет 4 канала, к которым можно подключать термосопротивление (Pt100, Pt1000, Cu50, R) для измерения температуры или измерения сопротивления. Тип канала настраивается в программном обеспечении. Каждый канал может быть настроен отдельно на конкретный тип датчика и поддерживает двухпроводные и трехпроводные датчики.



Входной сигнал каждого канала дискретизируется и рассчитывается АЦП для получения значения сопротивления, а затем соответствующее значение температуры вычисляется в соответствии с формулой температурного сопротивления для выбранного типа датчика и отправляется в область памяти AI через шину расширения для доступа пользовательской программе.

Различные формы сигналов имеют определенный диапазон измерения. Если измеренное значение превышает диапазон измерения, значение AI останется на соответствующем верхнем или нижнем предельном значении, и модуль расширения выдаст сигнал тревоги и отправит отчет о неисправности в CPU.

через шину расширения.

Неиспользуемые каналы рекомендуется соединить перемычкой как показано на схеме подключения и сконфигурировать в программном обеспечении как сопротивление, тогда эти неиспользуемые каналы не вызовут тревогу.

В следующей таблице показаны диапазон измерений и формат измеренных значений, где Т - измеренное значение температуры, а R - измеренное значение сопротивления.

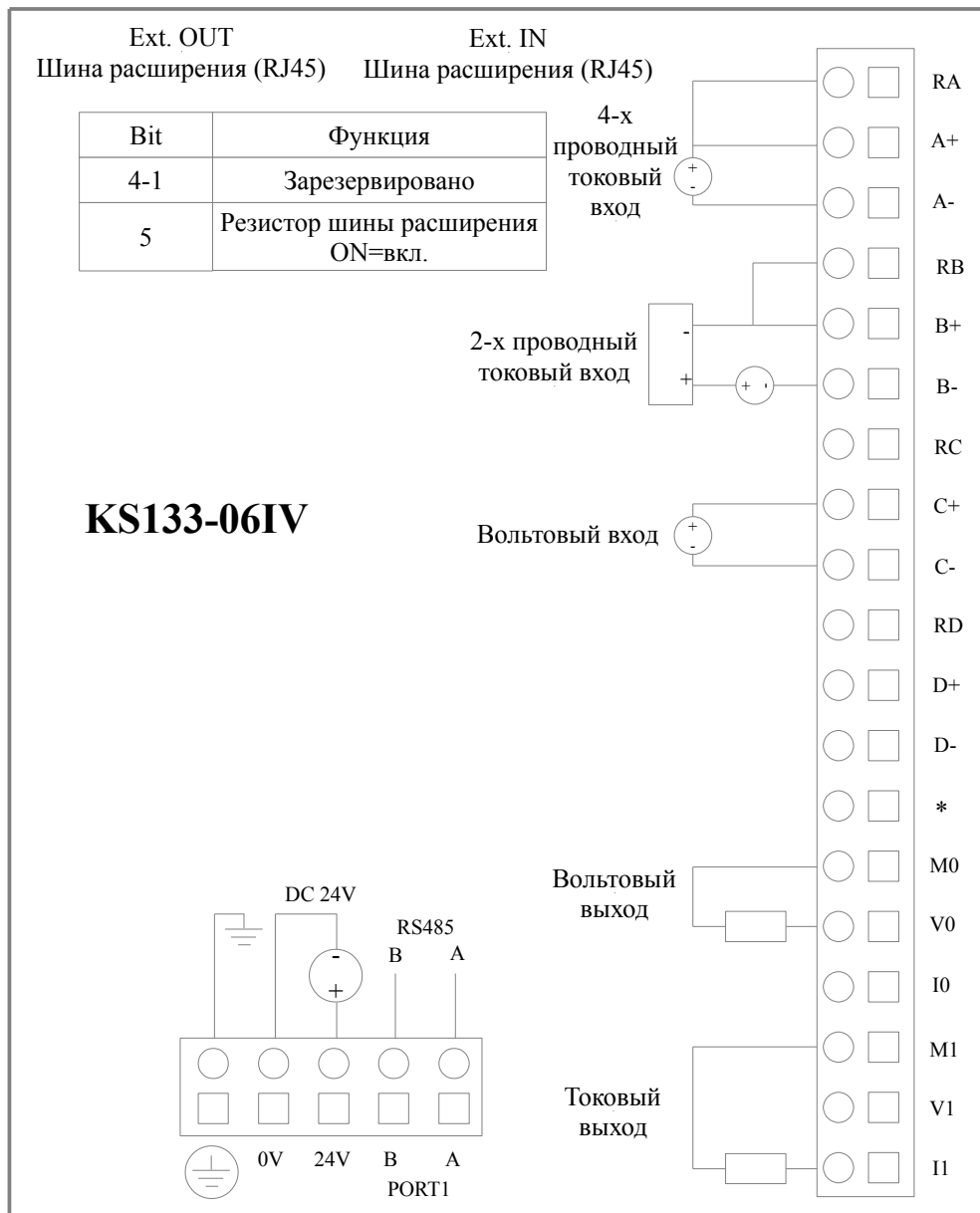
Тип сигнала	Диапазон измерения	Формат измеренных значений
Pt100	-200 ~ 850 °C	Tx10
Cu50	-50 ~ 150 °C	
Pt1000	-50 ~ 300 °C	
Сопротивление	0 ~ 200 Ом	Rx10

#### Технические характеристики KS131-04RD

Электрические параметры	
Номинальное напряжение питания	DC 24V , ≥100mA
Входы	4
Тип входа	Pt100, Cu50, Pt1000, Сопротивление
Тип подключения	Двухпроводный, трёхпроводный
Разрешение	24 Бит
Точность измерения	Температура : ± 0,6 °C, сопротивление : ± 1 Ом
Скорость преобразования (на канал)	Около 1раз / сек.
Размеры и вес	
Размеры (длина × ширина × высота)	100×84.5×25.4 мм
Вес	200г.

### 3.3.2 KS133-06IV

Этот модуль имеет 4 аналоговых входа и 2 аналоговых выхода. Измеряет и выводит стандартный сигнал напряжения и тока 4-20 мА, 1-5 В, 0-20 мА, 0-10 В. Все каналы в одном модуле расширения могут быть как токовыми так и вольтвыми.



#### Диапазоны измерений

##### \*Аналоговые входы

Входной сигнал в каждом канале будет измерять АЦП и счетчик. Результаты будут отправлены в область AI процессора с расширения CAN. Затем пользовательская программа может отобразить его.

Все типы сигналов имеют диапазон обнаружения. Если значение превышает диапазон, модули будут предупреждать, и LED индикатор будет включен. Тем временем модуль расширения отправит сигнал неисправности в CPU. Пожалуйста, подключите все входы, которые не используются, а также настройте тип сигнала **【0-20 мА】** или **【0-10 В】**, после чего эти каналы не будут предупреждать о неисправности.

Ниже представлен диапазон и значения. I - входной ток, V - входное напряжение.

Тип сигнала	Диапазон	Формат
4-20mA	3.92-20.4mA	Ix1000
0-20mA	0-20.4mA	
1-5V	0.96-5.1V	Vx1000
0-10V	0-10.2V	

**\*Аналоговые выходы**

Выход AQ будет отправлять модулям расширения аналоговый сигнал по шине, затем считать и изменять. После этого он будет выводиться из канала с помощью ЦАП. Диапазон выходного сигнала всех типов сигналов ограничен. Если выход превышает диапазон, выход будет поддерживать ограниченное значение сверху / снизу. Ниже представлена форма - это выходной диапазон и формат значений. I - реальный ток, V - реальное напряжение.

Тип сигнала	Выходной диапазон	Выходное значение
4-20mA	3.92-20.4mA	Ix1000
0-20mA	0-20.4mA	
1-5V	0.96-5.1V	Vx1000
0-10V	0-10.2V	

**Технические характеристики KS133-06IV**

Электрические параметры	
Номинальное напряжение питания	DC 24V , $\geq 100\text{mA}$
Аналоговые входы	4
Тип сигнала	4-20mA, 1-5V, 0-20mA, 0-10V
Разрешение	12 bit
Точность	0.3% от полной шкалы
Ограничение сигнала	Ток не превышает 24 мА, напряжение не превышает 12 В
Скорость изменения (каждый канал)	15 раз / S
Входное сопротивление	Токовый вход: $\leq 250\ \Omega$ Вольтовый вход: $> 4\ \text{M}\Omega$
Аналоговые выходы	2
Тип сигнала	4-20mA, 1-5V, 0-20mA, 0-10V
Разрешение	12 bit
Точность	0.3% от полной шкалы
Скорость изменения (каждый канал)	15 раз / S
Нагрузка	Токовый выход: Максимально 500 Ом Вольтовый выход: Минимально 10 кОм
Размеры и вес	
Размеры (длина × ширина × высота)	100×84.5×25.4 мм
Вес	200г.

## ГЛАВА 4 Программное обеспечение

На основе K5, KS использует то же программное обеспечение и инструкции Kincobuilder. Для большинства функций пользователи могут ссылаться на руководство K5 / K2. Основное различие - новые функции.

### 4.1 Протокол Modbus TCP для связи с сторонними устройствами

Области памяти, доступные мастеру Modbus TCP, классифицируются следующим образом:

Тип	Код функции	Соответствующая область памяти ПЛК
DO (Digital output, 0XXXX)	01, 05, 15	Q, M
DI (Digital input, 1XXXX)	02	I, M
AO (Analog output, 4XXXX)	03, 06, 16	AQ, V
AI (Analog input, 3XXXX)	04	AI, V
Запись об ошибке (16-битное целое число без знака)	03, 04	Область записи ошибок ПЛК

Максимальное количество регистров, которое может посетить одна инструкция:

1. Чтение бита: не более 1600 бит за один раз (200 байт) (код функции 01,02).
2. Запись бита: записать не более 800 бит за один раз (код функции 15).
3. Прочитать слово: прочитать не более 100 слов за один раз (код функции 03,04).
4. Запись слова: записать не более 100 слов за один раз (код функции 16).
5. Если диапазон памяти меньше указанного выше максимального значения, пользователь не может не только читать или записывать всю память, но и не может читать или записывать максимальный номер регистра, например, пользователь не может прочитать 90 слов в области AI (аналоговый ввод), потому что в этой области всего 32 слова.

#### 4.1.2 Регистры Modbus

Из-за различных областей памяти CPU с различными характеристиками, разрешенный диапазон доступа также ограничен. Для других спецификаций, Приложение А в разделе справки программного обеспечения для программирования KincoBuilder использует протокол Modbus RTU для связи. В некотором оборудовании регистры Modbus RTU начинаются с 1, поэтому к каждому значению в этом столбце следует добавить 1.

\*KS101M-04DX

Область	Диапазон	Тип	Соответствующий Modbus регистр
I	I0.0 --- I0.3	DI	0 --- 3
M	M0.0 --- M4095.7	DI/DO	320 -- 33087
V	VW0 --- VW16382	AI/AO	100 -- 8291

В дополнение к указанной выше области памяти, поддерживающей доступ по протоколу MODBUS, ПЛК серии K также предоставляет область памяти для записи ошибок, которую пользователи могут читать и просматривать через MODBUS. Для получения дополнительной информации см. Приложение D в разделе справки в программном обеспечении для программирования KincoBuilder, функция диагностики ошибок.

## 4.2 Высокоскоростной счетчик

KS имеет 4 высокоскоростных счетчика HSC0 ~ HSC3. Все HSC могут поддерживать до 200 кГц. Высокоскоростной счетчик поддерживает несколько режимов: однофазные, CW/CCW, фаза AB. Все высокоскоростные счетчики поддерживает максимум 32 PV и 32 "CV = PV" прерываний. PV может быть установлен как относительное или абсолютное значение.

### 4.2.1 Режимы работы и входы высокоскоростных счетчиков

Входные сигналы высокоскоростного счетчика включают в себя: счёт (входной импульс), направление, запуск и сброс. В различных режимах работы входные сигналы отличаются. Пожалуйста, смотрите ниже:

HSC0				
Режим	Описание	I0.1	I0.0	I0.5
0	Однофазный счетчик вверх / вниз с внутренним контролем направления: SM37.3	Счёт		
1			Сброс	
2			Сброс	Пуск
3	Однофазный счетчик вверх / вниз с внешним контролем направления	Счёт		Направление
4			Сброс	Направление
6	Двухфазный счетчик вверх / вниз с синхронизацией входов	Счёт вниз	Счёт вверх	
9	Счетчик с фазой A / B	Счёт A	Счёт B	

HSC1					
Режим	Описание	I0.4	I0.6	I0.3	I0.2
0	Однофазный счетчик вверх / вниз с внутренним контролем направления: SM47.3			Счёт	
1		Сброс			
2		Сброс	Пуск		
3	Однофазный счетчик вверх / вниз с внешним контролем направления			Счёт	Направление
4		Сброс			Направление
6	Двухфазный счетчик вверх / вниз с синхронизацией входов			Счёт вниз	Счёт вверх
7		Сброс			
9	Счетчик с фазой A / B			Счёт A	Счёт B
10		Сброс			

HSC2			
Режим	Описание	I0.4	I0.5
0	Однофазный счетчик вверх / вниз с внутренним контролем направления: SM57.3		Счёт

9	Счетчик с фазой А / В	Счёт А	Счёт В
---	-----------------------	--------	--------

HSC3			
Режим	Описание	I0.6	I0.7
0	Однофазный счетчик вверх / вниз с внутренним контролем направления: SM127.3		Счёт
9	Счетчик с фазой А / В	Счёт А	Счёт В

## 4.2.2 Байт управления и Байт состояния

### — Байт управления

В SM области, каждому высокоскоростному счетчику присваивается контрольный байт, чтобы сохранить свои данные конфигурации: одно управляющее слово (8 бит), текущее значение и предварительный набор (двойное число с 32 бит). Начальное значение текущей заданной величины. Если текущее значение записано в высокоскоростной счетчик, он будет начинать отсчет с этого значения. Пожалуйста, смотрите ниже:

HSC0	HSC1	HSC2	HSC3	Описание
SM37.0	SM47.0	SM57.0	SM127.0	Эффективный уровень сигнала сброса: 0 = высокий, 1 = низкий
SM37.1	SM47.1	SM57.1	SM127.1	Эффективный уровень сигнала старта: 0 = высокий, 1 = низкий
SM37.2	SM47.2	SM57.2	SM127.2	Скорость прямоугольного счетчика: 0 = 1х; 1 = 4х
SM37.3	SM47.3	SM57.3	SM127.3	Направление счета: 0 = уменьшение; 1 = увеличение
SM37.4	SM47.4	SM57.4	SM127.4	Записать направление счета в HSC? 0 = нет; 1 = да
SM37.5	SM47.5	SM57.5	SM127.5	Написать новое предустановленное значение в HSC? 0 = нет; 1 = да
SM37.6	SM47.6	SM57.6	SM127.6	Написать новое текущее значение HSC? 0 = нет; 1 = да
SM37.7	SM47.7	SM57.7	SM127.7	Разрешить этот высокоскоростной счетчик? 0 = нет; 1 = да
HSC0	HSC1	HSC2	HSC3	Описание
SMD38	SMD48	SMD58	SMD128	Текущее значение
SMD42	SMD52	SMD62	SMD132	Предустановленное значение

HSC0	HSC1	HSC2	HSC3	Описание
SM141.0	SM151.0	SM161.0	SM171.0	Использовать несколько предустановленных значений: 0 = нет; 1 = да
SM141.1	SM151.1	SM161.1	SM171.1	Предустановленное значение типа: 0 = абсолютное значение; 1 = относительное значение.
SM141.2	SM151.2	SM161.2	SM171.2	Предустановленное значение сравнения ("CV = PV") прерывания циклического выполнения: 0 = нет; 1 = да. <b>Примечание:</b> действительно только, когда заданное значение является относительной величиной.
SM141.3	SM151.3	SM161.3	SM171.3	Зарезервировано

SM141.4	SM151.4	SM161.4	SM171.4	Обновление нескольких сегментов PV и предустановленного значения: 0 = нет; 1 = да.
SM141.5	SM151.5	SM161.5	SM171.5	Сброс прерывания переменной: 0 = да; 1 = нет.
SM141.6	SM151.6	SM161.6	SM171.6	Зарезервировано
SM141.7	SM151.7	SM161.7	SM171.7	Зарезервировано
HSC0	HSC1	HSC2	HSC3	Описание
SMW142	SMW152	SMW162	SMW172	Начальное значение таблицы заданных значений (это смещение соответствующее VB0), это должно быть нечетным значением.

Обратите внимание, что не все управляющие биты управляющего байта подходят для всех режимов. Например, "Counting direction" и "Write counting direction in HSC" могут использоваться только в режиме 0,1 и 2 (Однофазный счетчик вверх / вниз с внутренним контролем направления), если режим работы с внешним управлением направления, то эти два бита будут игнорироваться. Управляющий байт, текущее значение и заданное значение = 0 по умолчанию после включения питания.

#### → Байт состояния

В SM области, каждый высокоскоростной счетчик имеет байт состояния, который показывает текущее состояние высокоскоростного счетчика.

HSC0	HSC1	HSC2	HSC3	Описание
SM36.0	SM46.0	SM56.0	SM126.0	Зарезервировано
SM36.1	SM46.1	SM56.1	SM126.1	Зарезервировано
SM36.2	SM46.2	SM56.2	SM126.2	Зарезервировано
SM36.3	SM46.3	SM56.3	SM126.3	Неисправность в нескольких табличных значениях PV: 0 = нет, 1 = да
SM36.4	SM46.4	SM56.4	SM126.4	Зарезервировано
SM36.5	SM46.5	SM56.5	SM126.5	Текущее направление счета: 0 = вниз; 1 = вверх
SM36.6	SM46.6	SM56.6	SM126.6	Текущее значение равно заданному значению: 0 = нет, 1 = да
SM36.7	SM46.7	SM56.7	SM126.7	Текущее значение больше, чем заданное значение: 0 = нет, 1 = да
HSC0	HSC1	HSC2	HSC3	Описание
SMB140	SMB150	SMB160	SMB170	Номер текущего сегмента PV (пуск с 0)

### 4.2.3 Установка заданного значения (значение PV)

KS поддерживает до 32 значений PV для каждого высокоскоростного счетчика, и поддерживает настройки значения PV, как относительного или абсолютного значения. Он поддерживает "CV = PV" прерывания циклического выполнения.

Следует принять HSC0 в качестве примера для описания функции значения PV и настройку.

#### → Как выбрать режим "multiple PV"

В контрольном байте каждого высокоскоростного счетчика есть один бит управления для включения нескольких предустановленных значений. В HSC0, этот контроль бит SM141.0.

Если SM141.0 = 0, он будет использовать одно значение PV, такой же, как PLC K5. SMD42 для нового значения PV, SM37.5 для обновления этого нового значения PV.



Если  $SM141.0 = 1$ , он будет использовать несколько значений PV. В этой ситуации, SM37.5 и SMD42 являются недействительными. Все значения PV будут в таблице PV (SMW142 для запуска адресов таблицы), SM141.4 определяет, использовать ли данные в таблице PV или нет. Если  $SM141.4 = 1$ , то HSC будет получать данные из таблицы PV, когда он включится. Если  $SM141.4 = 0$ , то HSC будет игнорировать данные в таблице PV и получать данные из последнего заданного значения.

#### — Таблица PV

При использовании таблицы PV, все значения PV получит из таблицы PV. Каждый HSC имеет одно управляющее слово, которое используется для установки начального адреса таблицы PV. При использовании **multiple PV**, все значения PV получит из таблицы PV. Начальным адресом PV таблицы является адрес V области, такой, как 301 (VB301). Формат таблицы PV выглядит следующим образом.

OFFSET <sup>(1)</sup>	Тип данных	Описание
0	BYTE	Количество PV
1	DINT	Первый PV
5	DINT	Второй PV
...	DINT	...

(1) Все значения смещения являются байтом смещения, связанные с таблицей.

(2) Когда он установлен в качестве относительного значения, то абсолютное значение данных PV должно быть больше, чем 1, или PLC будет рассматривать сегмент **multiple PV** конечным и подсчитает количество PV в соответствии с этим (с более высоким приоритетом, чем настройки количества PV).

Когда он установлен как абсолютное значение, разность между абсолютными значениями двух соседних PV должна быть больше, чем 1 или PLC будет рассматривать сегмент **multiple PV** конечным и подсчитает количество PV в соответствии с этим (с более высоким приоритетом, чем настройки количества PV).

(3) "CV = PV" прерывания должны выполняться последовательно, это означает, что после того, как счетчик достигнет первого PV и выполнит прерывание, то он будет сравнивать со вторым PV и так далее.

(4) PV должно быть установлено достаточно. Здесь принимает относительное значение, как, например, если счет положительный, PV должен быть больше 0, в противном случае прерывание "CV = PV" не будет выполняться. Если счет отрицательный, PV должно быть меньше 0, в противном случае "CV = PV" прерывание, также никогда не выполнится.

#### — Относительное значение и абсолютное значение

В байте управления каждого высокоскоростного счетчика, есть один бит управления, который используется для установки PV как относительного или абсолютного значения. Для HSC0 контроль бит SM141.1.

Если  $SM141.1 = 0$ , то PV является абсолютным значением. Когда значение счета равно PV, он будет выполнять "CV = PV" прерывание. Например, если установлено три значения PV, такие как 1000, 2000 и 3000, то когда значение счета достигнет 1000, он выполнит первое прерывание "CV = PV". Когда значение счета достигнет 2000, он выполнит второе прерывание "CV = PV" и так далее.

Если  $SM141.1 = 1$ , то PV является относительным значением. Если счетчик принимает текущее значение счета как опорное, в то время когда значение продолжает счёт равен PV, он выполнит прерывание "CV = PV". Например, если установлено три значения PV, такие как 10, 1000 и 1000, и текущее значение счета = 100 до начала HSC, затем, когда значение счёта достигнет 110, 1110 и 2110, он выполнит соответствующее прерывание "CV = PV".

#### — Циклическое выполнение прерывания "CV = PV"

Выполнение циклического прерывания "CV = PV" действительно только, когда PV устанавливается как относительное значение.

Если  $SM141.2 = 0$ , то прерывание "CV = PV" выполняется только один раз. Когда все прерывания закончат выполнение, то он остановится. Если нужно выполнить снова, то нужно изменить соответствующие

регистры и выполнить команду HSC снова.

Если SM141.2 = 1, то прерывание "CV = PV" будет выполняться циклически. Когда последнее прерывание PV завершит выполнение, PLC примет текущее значение счета в качестве эталона для расчета нового значения прерывания PV, затем он начнет сравнивать значение счета и выполнит прерывание "CV = PV" и так далее. Этот процесс будет выполнять циклически.

Например, установлено три значения PV, такие как 10, 1000 и 1000. И текущее значение счета = 100 до начала HSC, тогда значение для каждого прерывания в следующей таблице:

Текущее значение счёта	Время прерывания	Первое значение	Второе значение	Третье значение
100	1st	110	1110	2110
2110	2nd	2120	3120	4120
4120	3rd	4130	5130	6130
...	N	...	...	...

#### 4.2.4 Номер события "CV = PV"

Когда используется режим одного PV, HSC будет полностью совместим с K5 (включая номер события "CP = PV"). Когда используется режим **multiple PV**, HSC будет назначать новый номер события для 32 PV, как показано в следующей таблице.

Высокоскоростной счётчик	Номер прерывания	Описание
HSC0	64	"CV = PV" прерывание 1 PV
	65	"CV = PV" прерывание 2 PV
	...	... (+1)
	95	"CV = PV" прерывание 32 PV
HSC1	96	"CV = PV" прерывание 1 PV
	97	"CV = PV" прерывание 2 PV
	...	... (+1)
	127	"CV = PV" прерывание 32 PV
HSC2	128	"CV = PV" прерывание 1 PV
	129	"CV = PV" прерывание 2 PV
	...	... (+1)
	159	"CV = PV" прерывание 32 PV
HSC3	160	"CV = PV" прерывание 1 PV
	161	"CV = PV" прерывание 2 PV
	...	... (+1)
	191	"CV = PV" прерывание 32 PV

#### 4.2.5 Как использовать высокоскоростной счетчик

— **Метод 1: Используйте инструкции для программирования**

1) Настроить байт управления HSC и определить текущее значение (т.е. начальное значение) и установленное значение.

- 2) Использовать инструкцию HDEF для определения счета и его режима работы.
- 3) (Дополнительно) Используйте инструкцию ATCH для определения прерываний.
- 4) Используйте инструкцию HSC, чтобы запустить высокоскоростной счетчик.

#### — Метод 2: Использование «HSC Wizard»

В PLC K2 есть мастер конфигурации для высокоскоростного счетчика. Пользователи могут использовать его для настройки всех высокоскоростных счетчиков и не нужно программировать. После использования мастера для настройки HSC, пользователь также может использовать "метод 1", чтобы изменить параметры HSC. HSC Wizard показан на рисунке:

HSC Wizard

HSC:  Mode:  ☒ Enable HSC Start method:

Quadrature rate:  Reset signal level:  Start signal level:

Signal Input: Pulse: IO.1;

☒ Update direction New direction:   
☒ Update count value New count value:

☐ Enable external reset interrupt Interrupt routine:   
☐ Enable external direction-changed interrupt Interrupt routine:

PV and corresponding interrupts

☒ Enable multiple PVs Relationship between PVs:  ☐ Cyclic "CV=PV" interrupts

Multiple PVs settings

☒ Update PV and quantity Quantity:  Starting location of PV table (VB):

I...	Address	Value	Event...	Interrupt routine
1	%VD3010	100	64	(INT00) INT_0
2	%VD3014	200	65	(INT01) INT_1
3	%VD3018	300	66	(INT02) INT_2

Up  
Down  
Delete

Single PV settings (compatible with K5)

☐ Update preset value (PV) New PV:  ☐ Enable "CV=PV" interrupt Interrupt routine:

Apply OK Cancel Help

Как использовать **HSC Wizard**:

- 1) Выберите счетчик в [HSC]
- 2) Проверьте [Enable HSC], а затем продолжите следующие конфигурации.
- 3) Выберите режим счетчика в [Mode].
- 4) Выберите начальную режим в [Start method].

Есть два стартовых метода:

**"Using HSC instruction"**: если выбран этот метод, то нужно выполнить команду HSC, чтобы начать HSC. Перед выполнением инструкции HSC, не нужно настраивать регистры и выполнять инструкцию HDEF.

**"Run directly at PLC startup"**: При выборе этого метода, HSC начнется автоматически после включения ПЛК без выполнения каких-либо инструкций.

5) Если пользователю необходимо использовать режим **multiple PV**, поставьте галочку [Enable multiple PVs] и продолжайте настраивать все значения PV и связанные "Value" и "Interrupt subroutine". Если стоит галочка [Update PV and quantity], то его значение можно изменить в [Quantity] для изменения количества PV.

6) Если пользователю необходимо использовать режим одного PV, тогда поставьте галочку [Update preset value(PV)] в "Single PV settings" и измените значение PV и связанное прерывание подпрограммы.

7) Для других опций, пожалуйста, обратитесь к описаниям HSC.

### 4.3 Как использовать высокоскоростной импульсный выход

Kinco KS имеет 4 высокоскоростных импульсных выхода Q0.0, Q0.1, Q0.4 и Q0.5. При этом Q0.0, Q0.1 и Q0.4 поддерживают максимум 200 кГц, а Q0.5 поддерживает максимум 10кГц.

KS имеет выходной канал одного направления для каждого высокоскоростного выхода. KS обеспечивает управление одним направлением в области SM.

	Q0.0	Q0.1	Q0.4	Q0.5
Выход направления	Q0.2	Q0.3	Q0.6	Q0.7
Контроль направления	SM201.3	SM231.3	SM251.3	SM221.3

Сигнал направления: логический 0 — вращение вперед, логическая 1 — реверс.

Управление разрешением направления может запретить или разрешить выходной канал. Это наивысший приоритет. Если это запрещено, он не будет выводить сигнал управления направлением. Канал будет работать как обычный DO.

#### 4.3.1 Инструкция высокоскоростного импульсного выхода

KS имеет 3 типа инструкций для высокоскоростного импульсного выхода.

- 1) PLS: используется для вывода ПТО (один сегмент или несколько сегментов) и PWM.
- 2) Position control: имеет 5 инструкций, такие как PREL (относительное позиционирование), PABS (абсолютное позиционирование), PHOME (возврат в исходное положение), PJOG (Бег) и PSTOP (аварийный останов). Пользователь может использовать эти инструкции, чтобы достичь контроля позиционирования.

**Примечание:** При использовании команды управления позиции, частота выходного импульса должна быть не менее 125 Гц.

- 3) Инструкция PFLO\_F: имеет такие параметры, как входная частота (F), электронное передаточное отношение (NUM, DENOM), количество импульсов (COUNT) и так далее, эти параметры могут быть использованы в качестве переменной.

Частота импульсного выхода равна F кратной электронному передаточному отношению. Когда количество импульсов достигает значения COUNT, то выход остановится и установит бит DONE.

**Примечание:** При использовании этой команды, частота выходного импульса должна быть не менее 30 Гц.

### 4.3.2 Как использовать инструкцию PLS

PLS инструкция может выполнять выходную функцию PTO и PWM.

- PTO: Pulse Train Output (импульсный выход).
- PWM: Pulse-Width Modulation (широтноимпульсная модуляция).

— Описание

	Название	Использование	Группа	Подходит для
LD	PLS			K2 K5 KS
IL	PLS	PLS Q	U	

Операнд	Вход / выход	Тип данных	Описание
Q	Вход	INT	Константа (0,1,2 или 3)

Инструкция PLS используется для загрузки соответствующих конфигураций PTO / PWM из указанных регистров SM, а затем начинает вывод импульсов до тех пор, пока не закончится вывод импульсов. Импульсный выход определяется параметром Q, 0 означает Q0.0, 1 означает Q0.1, 2 означает Q0.4, 3 означает Q0.5.

**Примечание:** В программе пользователя, нужно только выполнить команду PLS, когда это требуется. Инструкция PLS выполняется по переднему фронту. Если на входе PLS всё время присутствует логическая единица, то она не будет выполняться.

#### ■ LD

Если EN равен 1, то PLS выполняется.

#### ■ IL

Если CR = 1, то PLS выполняется. Это не повлияет на значение CR.

#### 4.3.2.1 Функция высокоскоростного импульсного выхода Kinco-KS

Kinco-KS имеет 4 PTO / PWM генератора импульсов, которые могут быть использованы для выходов PTO / PWM. Поэтому, один генератор присваивается в Q0.0 и называется PWM0 или PTO0; второй присваивается в Q0.1 и называется PWM1 или PTO1, третий назначается в Q0.4 и называется PWM2 или PTO2. Четвёртый назначается в Q0.5 и называется PWM3 или PTO3.

Генераторы импульсов PTO / PWM и область отображения DO разделяются в адресе памяти Q0.0, Q0.1, Q0.4 и Q0.5. Когда в программе пользователя выполняются инструкции высокоскоростного импульсного выхода, то генератор PTO / PWM контролирует выход и запрещает нормальное использование этого выходного канала.

Некоторые регистры представлены в SM области для каждого генератора PTO / PWM. Когда пользователю необходимо использовать функцию импульсного выхода, он должен настроить эти регистры, а затем выполнить инструкцию PLS для реализации желаемой работы PTO / PWM.



**Примечание:** убедитесь, что не используете PTO и PWM функции, если Q0.0, Q0.1 и Q0.4 являются релейными выходами!



### → PWM

PWM обеспечивает непрерывный импульсный выход с переменным рабочим циклом, и вы можете контролировать время цикла и длительность импульса.

Единица времени цикла и времени длительности импульса микросекунды ( $\mu s$ ) или миллисекунды (ms). Максимальное значение времени цикла 65535. Если время длительности импульса больше, чем значение времени цикла, рабочий цикл задается на 100% автоматически и выход включен непрерывно. Если время длительности импульса равно 0, то рабочий цикл задается равным 0%, а выход выключен.

### → PTO

PTO обеспечивает прямоугольные импульсы (50% рабочего цикла) на выходе, и вы можете контролировать время цикла и количество выходных импульсов. Единица времени цикла микросекунды ( $\mu s$ ), или миллисекунды (ms). Максимальное значение времени цикла 65535. Диапазон числа импульсов 2 ~ 4,294,967,295. Если указанное число импульсов меньше 2, то Kinco K2 установит соответствующий бит ошибки и запретит выход.

PTO функция обеспечивает одиночный сегмент импульсов и множественный сегмент импульсов.

#### ■ Одиночный сегмент импульсов

В режиме одиночного сегмента импульсов он выполняет только выход последовательности импульсов один раз после выполнения инструкции PLS.

#### ■ Множественный сегмент импульсов

В режиме мульти сегментных импульсов, процессор автоматически считывает конфигурации каждого сегмента PTO из таблицы профиля, расположенной в области V и выполняет соответствующий сегмент PTO.

Длина каждого сегмента составляет 8 байт, в том числе значение времени цикла (16-бит, WORD), зарезервированное значение (не используется в настоящее время, 16-бит, INT), и значение количества импульсов (32-бит, DWORD). Вследствие этого, все выходные импульсы одни и те же, в том же сегменте. Он использует инструкцию PLS, чтобы начать много сегментные импульсы.

В этом режиме, начальный адрес таблицы хранится в SMW168 (соответствует PTO0), SMW178 (соответствует PTO1) и SMW268 (соответствует PTO2). Настройка базового времени SM67.3 (соответствует PTO0), SM77.3 (соответствует PTO1) и SM87.3 (соответствует PTO2). Единица измерения базового времени может быть микросекунды или миллисекунды. Все значения цикла в таблице должны использовать одну и ту же временную базу, и не могут быть изменены, когда профиль выполняется.

В следующей таблице описывается формат таблицы профиля.

Байт смещение <sup>1</sup>	Длина	Сегмент	Описание
0	8-bit		Количество сегментов (от 1 до 64)
1	16-bit	1	Начальное время цикла (от 2 до 65535 время от базы времени)
3	16-bit		Зарезервировано
5	32-bit		Количество импульсов (от 1 до 4294967295)
9	16-bit	2	Начальное время цикла (от 2 до 65535 время от базы времени)
11	16-bit		Зарезервировано
13	32-bit		Количество импульсов (от 1 до 4294967295)
...		...	...

(1) Все смещения в этой колонке по сравнению с исходной позицией в таблице профиля.

**Примечание:** начальная позиция в таблице профилей должна быть нечетным адрес в V области, например, VB3001.

## 4.3.2.2 Регистр PTO / PWM

Каждый генератор PTO / PWM снабжен некоторым регистром в области SM для хранения своей конфигурации, как показано в следующей таблице.

Q0.0	Q0.1	Q0.4	Q0.5	Описание
SM67.0	SM77.0	SM97.0	SM107.0	PTO / PWM. Следует ли обновлять время цикла: 0 = нет; 1 = да
SM67.1	SM77.1	SM97.1	SM107.1	PWM. Следует ли обновлять время ширины импульса: 0 = нет; 1 = да
SM67.2	SM77.2	SM97.2	SM107.2	PTO. Следует ли обновлять число импульсов: 0 = нет; 1 = да
SM67.3	SM77.3	SM97.3	SM107.3	PTO / PWM. Время базы: 0 = 1мкс; 1 = 1мс
SM67.4	SM77.4	SM97.4	SM107.4	PWM. Метод обновления: 0 = асинхронный; 1 = синхронный
SM67.5	SM77.5	SM97.5	SM107.5	PTO. Режим работы: 0 = одно сегментный; 1 = много сегментный
SM67.6	SM77.6	SM97.6	SM107.6	Выбор функции: 0 = PTO; 1 = PWM
SM67.7	SM77.7	SM97.7	SM107.7	PTO / PWM. Включение / отключение: 0 = отключить; 1 = включить

Q0.0	Q0.1	Q0.4	Q0.5	Описание
SMW68	SMW78	SMW98	SMW108	PTO / PWM. Время цикла, диапазон: 2 ~ 65535
SMW70	SMW80	SMW100	SMW110	PWM. Длительность импульса, диапазон: 0 ~ 65535
SMD72	SMD82	SMD102	SMD112	PTO. Количество импульсов, диапазон: 1 ~ 4,294,967,295
SMW168	SMW178	SMW218	SMW248	Исходное расположение таблицы профиля (байт смещения от V0), только для много сегментной работы PTO.

Все значения по умолчанию для управляющего байта, времени цикла и количества импульсов = 0. Что бы изменить конфигурацию PTO / PWM, во-первых нужно настроить соответствующие регистры управления, если это много сегментный импульс PTO, он также нуждается в настройке таблицы профиля, а затем выполнения команды PLS.

Каждый PTO / PWM генератор также имеет байт состояния в области SM, пользователь может получить информацию о состоянии PTO / PWM генератора от байта состояния, как показано в следующей таблице.

Q0.0	Q0.1	Q0.4	Q0.5	Описание
SM66.0	SM76.0	SM96.0	SM106.0	Зарезервировано
SM66.1	SM76.1	SM96.1	SM106.1	Зарезервировано
SM66.2	SM76.2	SM96.2	SM106.2	Зарезервировано
SM66.3	SM76.3	SM96.3	SM106.3	PWM в режиме ожидания: 0 = нет; 1 = да
SM66.4	SM76.4	SM96.4	SM106.4	Время цикла или число импульсов PTO неверное: 0 = нет; 1 = да <b>Примечание:</b> Время цикла и количество импульсов должны быть больше 1.
SM66.5	SM76.5	SM96.5	SM106.5	Профиль PTO прекращён в связи с командой пользователя: 0 = нет; 1 = да
SM66.6	SM76.6	SM96.6	SM106.6	Зарезервировано
SM66.7	SM76.7	SM96.7	SM106.7	PTO в режиме ожидания: 0 = нет; 1 = да

Бит ожидания PTO или бит ожидания PWM означает завершение PTO или PWM выхода.

### 4.3.2.3 Операции PTO

В качестве примера возьмём PTO0, что бы показать, как настраивать и эксплуатировать генератор PTO / PWM в программе пользователя.

Есть две процедуры для использования PTO: настройка соответствующих регистров управления и инициализации PTO. Выполнение команды PLS.

Используйте SM0.1 (первое сканирование бита памяти), чтобы вызвать подпрограмму, которая содержит инструкции инициализации. Поскольку SM0.1 используется, подпрограмма должна быть выполнена только один раз, и это снижает время сканирования CPU и обеспечивает лучшую структуру программы.

#### — Выполнение PTO (Одно сегментная работа)

1) Установите контрольный байт SMB67 в соответствии с желаемой работой.

Например, SMB67 = В # 16 # 85 означает:

- Включение PTO / PWM функции
- Выбор работы PTO
- Выбор 1 мкс как время базы
- Разрешение изменение числа импульсов и времени цикла.

2) Установите SMW68 в соответствии с желаемым временем цикла.

3) Установите SMD72 в соответствии с желаемым количеством импульсов.

4) (Необязательно) используйте ATCH для прикрепления событие PTO0-завершение (событие 28) к программе обработки прерываний, чтобы ответить в режиме реального времени на событие PTO0-завершение.

5) Выполните команду PLS, чтобы настроить PTO0 и запустить его.

#### — Изменение времени цикла PTO (Одно сегментная работа)

Выполните следующие шаги, чтобы изменить время цикла PTO.

1) Установите контрольный байт SMB67 в соответствии с желаемой работой.

Например, SMB67 = В # 16 # 81 означает:

- Включение PTO / PWM функции
- Выбор операции PTO
- Выбор 1 мкс как время базы
- Разрешение обновления значения времени цикла.

2) Установите SMW68 в соответствии с желаемым временем цикла.

3) Выполните команду PLS, чтобы настроить PTO0 и запустить его, после этого новый PTO с обновленным временем цикла начнёт генерироваться.

#### — Изменение числа импульсов PTO (Одно сегментная работа)

Выполните следующие действия, чтобы изменить количество импульсов PTO:

1) Установите контрольный байт SMB67 в соответствии с желаемой работой.

Например, SMB67 = В # 16 # 84 означает:

- Включение PTO / PWM функции
- Выбор операции PTO
- Выбор 1 мкс как время базы
- Разрешение изменения числа импульсов

2) Установите SMD72 в соответствии с желаемым количеством импульсов.

3) Выполните команду PLS, чтобы настроить PTO0 и запустить его, после этого новый PTO с обновленным количеством импульсов начнёт генерироваться.



**→ Выполнение PTO (Много сегментная работа)**

1) Установите контрольный байт SMB67 в соответствии с желаемой работой.

Например, SMB67 = В # 16 # A0 означает:

- Включение PTO / PWM функции
- Выбор операции PTO
- Выбор много сегментной работы
- Выбор 1 мкс как время базы

2) Установите нечетное число в качестве исходного положения таблицы профиля в SMW168.

3) Используйте V область, чтобы настроить таблицу профиля.

4) (Необязательно) используйте ATCH для прикрепления событие PTO0-завершение (событие 28) к программе обработки прерываний, чтобы ответить в режиме реального времени на событие PTO0-завершение.

5) Выполните команду PLS, чтобы настроить PTO0 и запустить его.

**4.3.2.4 Операции PWM**

В качестве примера возьмём PWM0, что бы показать, как настраивать и эксплуатировать генератор PTO / PWM в программе пользователя.

Есть две процедуры для использования PWM: настройка соответствующих регистров управления и инициализации PWM. Выполнение команды PLS.

Используйте SM0.1 (первое сканирование бита памяти), чтобы вызвать подпрограмму, которая содержит инструкции инициализации. Поскольку SM0.1 используется, подпрограмма должна быть выполнена только один раз, и это снижает время сканирования CPU и обеспечивает лучшую структуру программы.

**→ Выполнение PWM**

1) Установите контрольный байт SMB67 в соответствии с желаемой работой.

Например, SMB67 = В # 16 # D3 означает:

- Включение PTO / PWM функции
- Выбор операцию PWM
- Выбор 1 мкс как время базы
- Разрешение обновления значения ширины импульса и времени значение цикла

2) Установите SMW68 в соответствии с желаемым временем цикла.

3) Установите SMW70 в соответствии с требуемой шириной импульса.

4) Выполните команду PLS, чтобы настроить PWM0 и запустить его.

**→ Изменение длительности импульса для PWM выхода**

Далее описываются шаги для изменения длительности выходного импульса PWM.

1) Установите контрольный байт SMB67 в соответствии с желаемой работой.

Например, SMB67 = В # 16 # D2 означает:

- Включение PTO / PWM функции
- Выбор операции PWM
- Выбор 1 мкс как время базы
- Разрешить обновления значения ширины импульса и времени значения цикла

2) Установите SMW70 в соответствии с требуемой шириной импульса.

3) Выполните команду PLS, чтобы настроить PWM0 и запустить его.

### 4.3.3 Как использовать команды управления позицией

#### 4.3.3.1 Как изменить текущее значение команды управления позицией

##### — Регистры управления и регистры состояния

Для команд управления позицией Kinco-KS определяет управляющий байт для каждого высокоскоростного выходного канала для хранения свои настроек. Кроме того, он назначает регистр текущего значения (DINT) для хранения количества импульсов, которое в настоящее время выводятся (это значение будет увеличиваться при счёте вперед и снижаться при счёте назад). В следующей таблице описаны байт управления и текущее значение.

Q0.0	Q0.1	Q0.4	Q0.5	Описание
SMD212	SMD242	SMD262	SMD226	Только для чтения. Текущее значение (увеличиваться при счёте вперед и снижаться при счёте назад). Означает количество импульсов, которое уже выводится.
SMD208	SMD238	SDM258	SDM222	Чтение / запись. Новое текущее значение. Используется для изменения текущего значение с определенным битом управления.
Q0.0	Q0.1	Q0.4	Q0.5	Описание
SM201.7	SM231.7	SM251.7	SM221.7	Чтение / запись. Бит аварийного останова. Если этот бит равен 1, никаких инструкций контроля положения не могут быть выполнены. При выполнении инструкции PSTOP, этот бит устанавливается = 1 автоматически, и он должен быть сброшен в программе.
SM201.6	SM231.6	SM251.6	SM221.6	Чтение / запись. Сброс текущего значения: 1 = очищать текущее значение 0 = поддерживать текущее значение
SM201.5	SM231.5	SM251.5	SM221.5	Зарезервировано
SM201.4	SM231.4	SM251.4	SM221.4	Чтение / запись. Используется для изменения текущее значения: 1 = изменение текущего значения 0 = поддерживать текущее значение
SM201.3	SM231.3	SM251.3	SM221.3	Чтение / запись. Бит управление направлением: 1 = отключение выходного канала направления, он будет использоваться в качестве обычного выхода 0 = включить выходной канал направления
SM201.2 ~ SM201.0	SM231.2 ~ SM231.0	SM251.2 ~ SM251.0	SM221.2 ~ SM221.0	Зарезервировано

##### — Как изменить текущее значение

Каждый высокоскоростной выходной канал имеет один регистр для текущего значения, SMD212, SMD242 и SMD262, SMD226. Выведенное количество импульсов хранится в этих регистрах. Регистры текущего значения только для чтения, если пользователю нужно изменить текущее значение, он может использовать следующие методы:

### ■ Метод 1

Использовать бит сброса для очистки текущего значения.

Бит сброса для трёх выходных каналов SM201.6, SM231.6 и SM251.6, SM221.6.

Когда бит сброса = 1, ПЛК установит текущее значение = 0. Таким образом, требуется только одно сканирование для активации бита сброса. Когда нужно использовать этот бит, старайтесь избегать, чтобы этот бит всегда был = 1, а также старайтесь не устанавливать этот бит во время выполнения команды Position Control (PHOME, PREL, PABS, JOG и PFLO\_F), в противном случае значение счетчика может быть неправильным.

Возьмём для примера канал 0 для описания, как сбросить текущее значение.

(\* Network \* 0)

(\* Основано на сигнале возврата в исходное положение, когда он перемещается в исходное положение, требуется очистить текущее значение \*)

LD %SM0.0

PHOME 0, %M0.0, %M0.1, %M0.2, %VW0, %VW2, %VW4, %VD6, %VW10, %M0.4, %M0.5, %MB1

(\* Network \* 1)

(\* После окончания PHOME, используйте бит завершения "DONE", чтобы очистить текущее значение \*)

LD %M0.4

R\_TRIG

ST %SM201.6

### ■ Метод 2

Изменение текущего значения с помощью следующих регистров.

Q0.0	Q0.1	Q0.4	Q0.5	Описание
SMD208	SMD238	SMD258	SMD222	Чтение / запись. Новое текущее значение. Используется для изменения текущего значения с определенным битом управления.
SM201.4	SM231.4	SM251.4	SM221.4	Чтение / запись. Используется для изменения текущего значения: 1 = изменение текущего значения 0 = поддерживать текущее значение

Возьмём для примера канал 0 для описания метода. Если SM201.4 = 0, то он будет поддерживать текущее значение SMD212. Если SM201.4 = 1, то он будет перемещать значение SMD208 в SMD212. Когда нужно использовать этот бит, старайтесь избегать, чтобы этот бит всегда был = 1, а также старайтесь не устанавливать этот бит во время выполнения команды Position Control (PHOME, PREL, PABS, JOG и PFLO\_F), в противном случае значение счетчика может быть неправильным.

Возьмём для примера канал 0 для описания, как изменить текущее значение:

(\* Network 0 \*)

(\* Основано на сигнале возврата в исходное положение, когда он перемещается в исходное положение, требует установить текущее значение = 100. \*)

LD %SM0.0

PHOME 0, %M0.0, M0.1, %M0.2, %адресу VW0, %VW2, %VW4, %VD6, %VW10, %M0.4, M0.5, %MB1

(\* Network 1 \*)

(\* После окончания PHOME, используйте бит завершения "DONE", что бы изменить текущее значение. \*)

LD %M0.4

R\_TRIG

MOVE DI # 100, %SMD208

ST %SM201.4

### 4.3.3.2 Можно ли изменить максимальную выходную частоту, когда выполняется инструкция управления положением?

PREL (относительное положение) и PABS (абсолютное положение) не изменят максимальную выходную частоту, когда они выполняются. Они будут считывать параметры Минимальная частота, Максимальная частота и Время разгона / торможения, когда он запускается, и вычисляет подходящее ускорение / торможение в соответствии со значением этих параметров, затем он начнет выход пульсов. Во время выхода импульсов, PREL и PABS не будут читать новые параметры, поэтому, изменение этих параметров не влияет на импульсный выход.

PJOG (бег) будет считывать входную частоту импульса (MAXF) все время, когда он выполняется, и регулировать частоту выходного импульса в соответствии с новой настройкой частоты.

PHOME (возврат в исходное) будет считывать максимальную частоту (MAXF) все время, когда он работает на максимальной частоте, но не найдёт сигнала возврата в исходное положение, и вычислит ускорение или торможение автоматически в соответствии с новой настройкой частоты, затем он будет ускорять или замедлять выходной импульс с новой частотой.

## 4.4 Использование CAN шины

KS105C1-16DT имеет 1 порт CAN, CAN2

KS105C2-16DT имеет 2 порта CAN, CAN1 и CAN2.

CAN2 поддерживает протокол CANopen master и протокол free. CAN1 поддерживает free протокол.

Основная функция CANopen KS такая же, как K5.

Для free протокола используйте инструкции по обмену данными ((CAN\_INIT, CAN\_WRITE, CAN\_READ, CAN\_RX), параметры CH, 0 — CAN1, 1 — CAN2.

Модель	CAN порт	Протокол расширения	Свободный протокол	CANOpen master	Протокол Motion Control	CANOpen slave
KS105C1	CAN	---	Поддерживает, можно выбрать только один из них.			
KS105C2 KS101M	CAN1	До 14 расширений	Да	---	---	---
	CAN2	---	Поддерживает, можно выбрать только один из них.			---

**Примечание:** Когда к интерфейсу CAN1 подключен модуль расширения, функция связи CAN недействительна, то есть можно выбрать только один, протокол расширения или функцию связи CAN.

### 4.4.1 Подключение

KS105C1-16DT имеет один интерфейс CAN, который напрямую подключается к двум разъемам RJ45, что позволяет пользователю использовать один из них. См. схему подключения в 2.3.

KS105C2-16DT / KS101M-04DX имеет два интерфейса CAN, которые разделены на два разъема RJ45. См. схему подключения в 2.3. На корпусе ПЛК CAN-шины имеется DIP-переключатель, позволяющий, включить клеммное сопротивление.

**Примечание:** Когда несколько модулей объединены в сеть, соединительный кабель необходимо подключить с помощью прямого сетевого кабеля, а не перекрестного сетевого кабеля.

#### 4.4.2 Функция шины расширения

Коммуникационный порт CAN поддерживает протокол шины расширения. Если модуль расширения настроен в **[Hardware Configuration]** пользовательского проекта, интерфейс CAN будет работать как интерфейс шины расширения и в пользовательской программе разрешена только команда свободного обмена данными.

ПЛК позволяет подключать до 14 модулей расширения. Каждый модуль расширения допускает распределенную установку. Общая длина коммуникационного кабеля между ПЛК и последним модулем расширения не должна превышать 30 метров. При использовании шины расширения рекомендуется добавить оконечное сопротивление на ПЛК и на последнем модуле расширения, чтобы избежать отражения сигнала и повысить стабильность связи. Кроме того, при использовании распределенной установки на большие расстояния для удлиненного коммуникационного кабеля рекомендуется использовать экранированную витую пару, несимметричный экран должен быть хорошо заземлен, коммуникационный кабель должен находиться вдали от источников сильных помех и различных высоковольтных линий. В заводских настройках по умолчанию ПЛК автоматически назначает уникальный идентификатор каждому модулю расширения и настраивает различные параметры при включении. Следовательно, требуется, чтобы ПЛК и все модули расширения были включены одновременно, или все модули расширения включены вперед ПЛК. Включите питание, иначе это может вызвать ошибку выполнения программы. Чтобы упростить распределенное использование, CPU предоставляет инструкцию EX\_ADDR, и пользователь может изменять указанную выше конфигурацию по умолчанию, вызывая эту инструкцию, тем самым делая использование модуля расширения более гибким.

##### Как использовать инструкцию EX\_ADDR для модуля расширения.

В практических приложениях, если модуль расширения находится далеко от модуля CPU или установлен на другом устройстве, последовательность включения по умолчанию не может быть гарантирована. В этом случае пользователь может использовать инструкцию EX\_ADDR для изменения заводской конфигурации по умолчанию, как описано ниже, чтобы каждый модуль расширения мог быть включен или выключен в любое время, не вызывая программную ошибку ПЛК.

- 1) В пользовательском проекте в **[Hardware Configuration]** добавьте каждый модуль расширения в требуемом порядке и настройте его в соответствии с фактическими требованиями, а также вызовите инструкцию EX\_ADDR в программе.
- 2) Подключите CPU и все модули расширения в соответствии с порядком **[Hardware Configuration]**, а затем включите питание.
- 3) Загрузите пользовательский проект в CPU. После начала нормальной работы CPU измените значение параметра инструкции EX\_ADDR на 181 (десятичное), а затем выполните команду EX\_ADDR один раз. После успешного выполнения команды каждый модуль расширения автоматически сохранит свой собственный ID и различные параметры (такие как форма сигнала, метод фильтрации и т.д.).
- 4) Выключите питание ПЛК и модулей расширения. Затем можно установить модуль расширения в желаемом месте, сохраняя порядок модулей (начиная с CPU) таким же, как в **[Hardware Configuration]**. После этого при включении модуля расширения он автоматически считывает сохраненные данные и автоматически переходит в рабочее состояние. Его не нужно настраивать с помощью CPU, поэтому его можно включить или выключить в любое время независимо от CPU.
- 5) Если необходимо восстановить заводскую последовательность включения по умолчанию, измените значение параметра инструкции EX\_ADDR в программе на 99 (десятичное), а затем выполните инструкцию EX\_ADDR один раз. После успешного выполнения инструкции каждый модуль расширения очищает сохраненный ID и параметры канала. После включения питания он будет ждать, пока CPU автоматически назначит ID и настроит параметры.



### 4.4.3 Функция Kinco Motion Control

Функция **Motion Control** используется для управления движением продуктов Kinco (сервоприводы и шаговые приводы) с интерфейсом CAN. Основанный на протоколе CANOpen, он инкапсулирует детали связи CANOpen драйвера и предоставляет пользователю набор инструкций по управлению движением и соответствующие инструменты конфигурации сети в сочетании с фактическими требованиями приложения. Эта функция проста в использовании, и пользователи могут легко связываться с приводом и выполнять управление позиционированием, даже если они не знакомы с деталями протокола CANOpen. Эта функция может управлять до 32 продуктами движения. В практических приложениях пользователь может определить фактическое количество подключенных устройств в соответствии с требуемым программным пространством, скоростью загрузки сети и т.п.

Эта функция поддерживает загрузку/выгрузку параметров, блокировки оси двигателя, освобождения вала, поиск нуля, толчковый режим (режим скорости), абсолютное позиционирование, относительное позиционирование и т. д. Но эта функция не поддерживает режим крутящего момента и следование ведущему-ведомому. Кроме того, эта функция может использоваться для всех сторонних продуктов управления движением, которые поддерживают стандартный протокол CANOpen. **Перед использованием проконсультируйтесь с техническим специалистом.**

Пользователь использует функцию **Motion Control** Kinco следующим образом:

- 1) В пользовательском проекте войдите в окно **[Kinco Motion Control Wizard]**, чтобы завершить настройку сети.
- 2) Вызовите инструкцию управления движением, чтобы запрограммировать ее в соответствии с фактическими потребностями.
- 3) Когда проект загружен в ПЛК, он будет работать как главная станция, управлять связью всей сети и выполнять программу управления позиционированием.

#### Kinco Motion Control Network Configuration

Функция управления движением Kinco использует протокол CANOpen, ПЛК действует как главная станция, а каждый драйвер действует как подчиненный. Перед вызовом команды пользователь должен сначала настроить фактическую используемую сеть CANOpen. В **[Project Manager]** программного обеспечения Kincobuilder дважды щелкните узел **[Kinco Motion Control Wizard]**, чтобы войти в окно конфигурации и завершить настройку сети.

Окно разделено на три части: древовидный список сетевых узлов, параметры мастера и параметры оси (подчиненного).

#### Работа с деревом сетевых узлов

В дереве сетевых узлов корневым узлом является **[CANOpen Master]**, а каждый из следующих дочерних узлов является осью (ведомым) в сети. Ниже представлены четыре кнопки **[Add]**, **[Delete]**, **[Copy]** и **[Paste]**, а программное обеспечение также предоставляет соответствующие сочетания клавиш и функции контекстного меню. Пользователи могут использовать эти функции для работы с деревом сетевых узлов.

#### Добавить новую ось

Щелкните кнопку **[Add]** или щелкните правой кнопкой мыши любой узел и выполните команду добавления. Или используйте сочетание клавиш ALT + A. Все оси, добавленные с помощью трех вышеуказанных методов, начинаются с параметров по умолчанию.

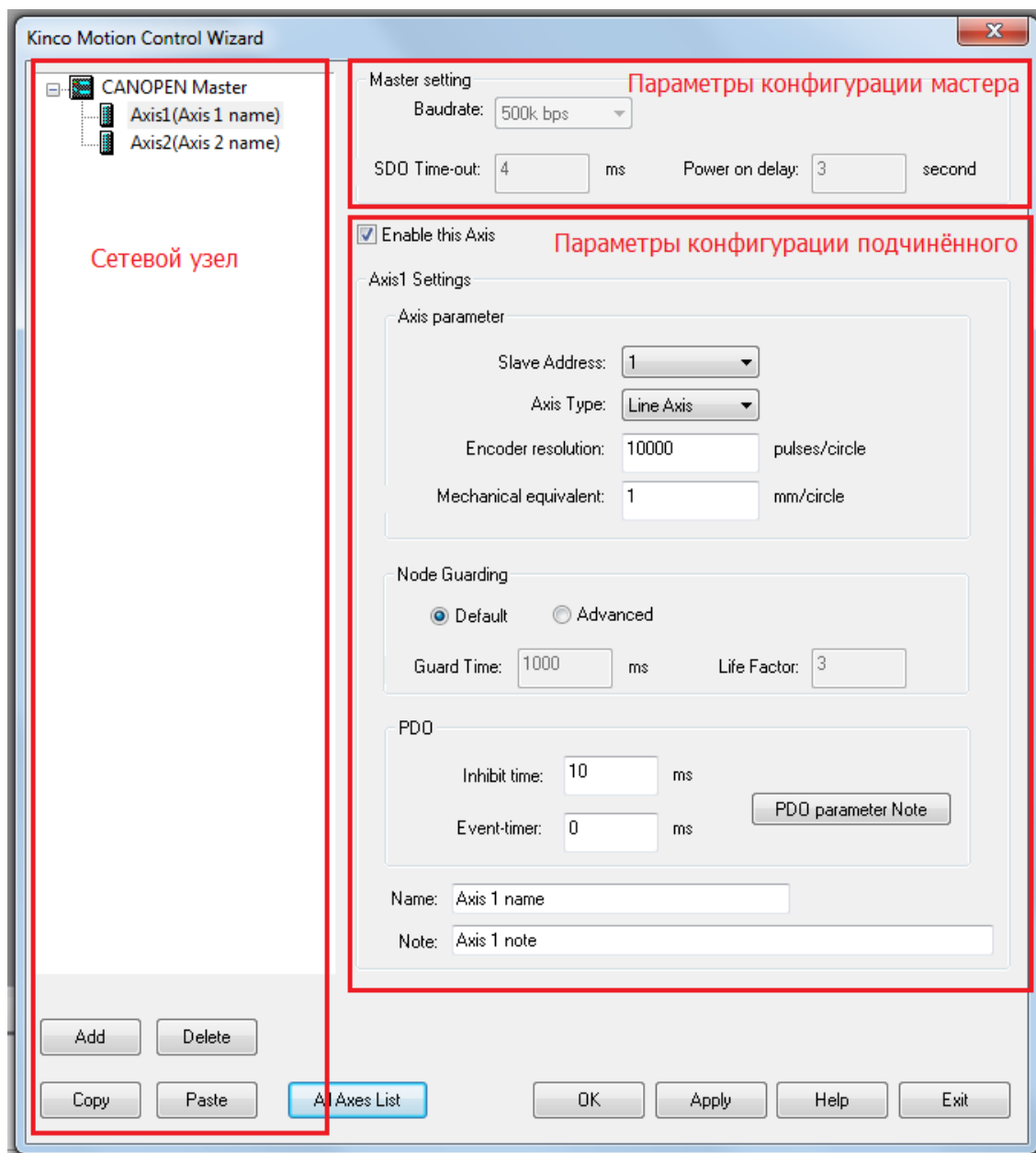
#### Копировать / вставить

Пользователь может скопировать существующую ось, а затем вставить ее в сеть для создания новой оси. Новая ось имеет те же параметры, что и скопированная ось, за исключением номера оси (адреса подчиненной станции). Это удобно для проектов, где все оси в сети работают одинаково. Щелкните ось в дереве, чтобы выбрать ее, затем нажмите кнопку копирования или используйте Ctrl + C. Или щелкните правой кнопкой мыши по оси и выполните команду меню «Копировать». Вы можете продублировать эту ось любым из этих способов. После копирования нажмите кнопку «вставить», или используйте сочетание клавиш Ctrl + V, или щелкните правой кнопкой мыши любую ось и выполните команду меню вставки, чтобы

создать новую ось в сети.

### Удалить

Вы можете удалить ось, щелкнув ось, чтобы выбрать ее, затем нажмите кнопку **[Delete]**. Так же можете удалить ось, щелкнув ее правой кнопкой мыши и выполнив команду меню удаления.



### Параметры конфигурации мастера

Щелкните узел [CANOpen master], все параметры главной станции станут доступными для изменения, а все параметры оси (подчиненной станции) станут серыми и неизменяемыми.

- **[Baudrate]**: Выберите скорость передачи, используемую главной станцией. Обратите внимание, что скорости передачи должны быть согласованы на всех узлах (главном и подчиненном) в сети.
- **[SDO Time-out]**: Время ожидания тайм-аута после того, как ПЛК ведущей станции отправит сообщение запроса SDO, если по истечении этого времени не будет получено ответное сообщение от соответствующей ведомой станции, будет сообщено об ошибке тайм-аута. При выборе другой скорости передачи программное обеспечение автоматически рекомендует тайм-аут SDO, который пользователь может изменить.

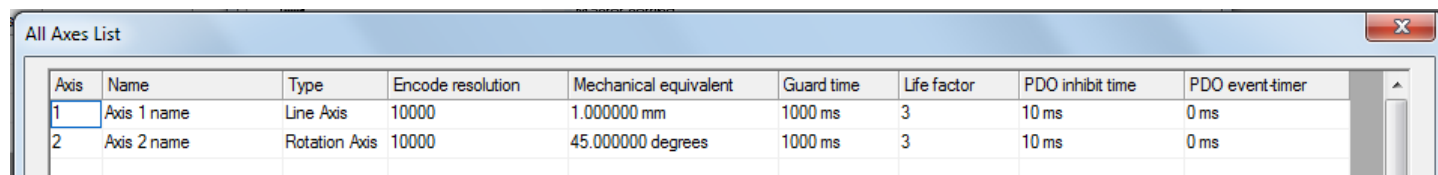
### Ось (параметры конфигурации подчинённого)

Когда вы щелкаете по узлу оси, все параметры оси становятся доступными для изменения, а все параметры первичной станции будут серыми и неизменяемыми.

- **[Slave Address]**: Ось адреса ведомой станции CANOpen. Последовательное присвоение номера станции должно начинаться с 1.
- **[Axis type]**: В зависимости от функции вала пользователь может выбрать прямой или вращающийся вал.
- **[Encoder resolution]**: Разрешающая способность энкодера вала или шагового драйвера, то есть количество импульсов энкодера за один оборот.
- **[Mechanical equivalent]**: Для каждого вращения вала двигателя длина (мм) или угол перемещения механической нагрузки.
- **[Node Guarding]**: Установите время защиты узла оси. Пользователи могут использовать значение по умолчанию или выбрать «дополнительно», чтобы изменить его.
- **[PDO]**: PDO автоматически устанавливается в ПЛК для каждой оси для передачи такой информации, как положение, скорость и состояние. Поскольку положение и скорость оси меняются быстро, PDO отправляется очень часто, поэтому для PDO необходимо установить запрет на время. Пользователь может использовать значения по умолчанию или изменять их.

### Другое

- **[OK]** : Сохранить параметр и выйти.
- **[Cancle]** : Не сохранять и выйти
- **[Apply]** : Сохраняет параметры, настроенные текущим параметром.
- **[All Axes List]** : Контрольный список осей предназначен для облегчения проверки всех настроенных и включенных параметров конфигурации осей для проверки.



Axis	Name	Type	Encode resolution	Mechanical equivalent	Guard time	Life factor	PDO inhibit time	PDO event-timer
1	Axis 1 name	Line Axis	10000	1.000000 mm	1000 ms	3	10 ms	0 ms
2	Axis 2 name	Rotation Axis	10000	45.000000 degrees	1000 ms	3	10 ms	0 ms

## 4.4.4 Функция CANOpen Master

CANOpen обладает такими преимуществами, как хорошая открытость, высокая надежность, хорошая производительность в реальном времени, сильная защита от помех и низкая стоимость. Это широко используемая полевая шина в промышленном управлении, и в настоящее время она находит все более широкое применение.

Уровень приложений и спецификация связи CANOpen (CiA DS301) является ядром протокола CANOpen и применяется ко всем устройствам CANOpen. Различные объекты связи CANOpen определены в DS301, а службы и протоколы для этих объектов подробно описаны. Чтобы облегчить приложение пользователя, мы представим несколько ключевых объектов и их протоколы связи.

### Network Management Tool (NMT)

Сетевое управление (NMT) - это модель «ведущий-ведомый» для устройств CANOpen. Служба NMT может инициализировать, запускать, контролировать, сбрасывать или останавливать устройства CANOpen. В сети должна быть одна мастер-станция NMT, которая контролирует всю сеть, функцию класса управления сетью (NMT). Вот несколько распространенных сервисов NMT.

### NMT Node Control

Мастер-станция NMT управляет состоянием NMT (включая остановку, предварительную работу, работу и



инициализацию) каждой подчиненной станции через сообщение управления узлом NMT. Устройства ведомой станции должны поддерживать службы управления узлами NMT.

Формат управляющего сообщения узла NMT следующий:

COB-ID	Byte 0	Byte 1
0x000	CS (указатель команд)	Node ID

Node ID: ID ведомого, если он равен 0, это означает, что все ведомые станции в сети должны выполнить эту команду.

CS спецификатор команды:

- 1 означает запуск целевого узла;
- 2 означает остановить целевой узел;
- 128 означает целевой узел готов;
- 129 означает сброс целевого узла;
- 130 означает целевой узел сбрасывает параметры связи.

### NMT Error Control

Служба контроля ошибок, используемая для обнаружения сбоев сети, включает защиту узлов (Node Guarding) и контрольные сообщения (Heartbeat). На практике вы должны выбрать метод контроля ошибок для узла. Кстати, служба Heartbeat добавлена в более позднюю версию DS301 и рекомендована CiA.

### NMT Node Guarding

NMT мастер отправляет удаленные кадры (без данных):

COB-ID
0x700 + Node ID

NMT ведомый возвращает кадр:

COB-ID	Byte 0
0x700 + Node ID	Бит 7: триггерный бит, должен чередоваться «0» или «1» в каждом ответе защиты узла. Бит 0-6: комбинация значения представляет состояние ведомой станции. Где 0 означает загрузку, а 4 означает остановлено; 5 означает работающий; 127 означает предработавший.

### NMT Heartbeat

Если узел сконфигурирован на отправку контрольных сообщений, он периодически посылает их, а другой узел или узлы в сети принимают эти сообщения для обработки. Как правило, ведущая станция принимает контрольные сообщения, а другие подчиненные станции отправляют их.

Формат контрольного сообщения выглядит следующим образом:

COB-ID	Byte 0
0x700 + Node ID	Значение состояния этого узла. Где 0 означает загрузку, а 4 означает остановлено; 5 означает работающий; 127 означает предработавший.

### SDO (Service Data Object)

Связь SDO основана на модели клиент-сервер.

Используя индекс и субиндекс, SDO позволяет одному устройству CANOpen (в качестве клиента) напрямую обращаться к объектам в словаре объектов другого устройства CANOpen (в качестве сервера). Как правило,

первичная станция действует как клиент.

SDO имеет два режима транспортировки: ускоренный тип, с одновременной передачей до четырех байтов данных; сегментированный тип, позволяющий сегментировать более 4 байтов данных.

Ниже приведено краткое описание формата сообщения транспортного режима ускоренного типа:

SDO кадр запроса, Клиент -> Сервер :

COB-ID	Byte 0	Byte 1-2	Byte 3	Byte 4-7
0x600 + Node ID	SDO CS	Index	Sub-Index	Data

Кадр ответа, Сервер -> Клиент:

COB-ID	Byte 0	Byte 1-2	Byte 3	Byte 4-7
0x580 + Node ID	SDO CS	Index	Sub-Index	Data

### PDO (Process Data Object)

PDO для передачи данных в реальном времени, сообщение PDO содержит до 8 байт данных. Коммуникация PDO основана на модели производитель-потребитель. Что касается отправки или приема данных, PDO делится на отправляющий PDO (TPDO) и принимающий PDO (RPDO). Производители поддерживают TPDO, а потребители поддерживают RPDO. Связь PDO не имеет протокола, который гласит, что содержимое в сообщении PDO, предварительно определено. Во время конфигурации сети пользователь определяет COB-ID каждого PDO и отображенных в нем объектов, так что как производитель, так и потребитель могут знать содержимое соответствующего PDO и анализировать сообщение соответствующим образом. Каждый PDO описывается в словаре объектов с помощью параметров связи и отображения.

Параметры связи PDO описаны ниже.

#### COB-ID

Указывает COB-ID, используемый PDO.

#### Transmission type

Указывает, как PDO иницируется для отправки (или получения). Это 8-битное целое число без знака.

Типы передачи делятся на следующие категории:

- Синхронизация: иницирует отправку (или получение) на основе счетчика объекта SYNC.

Если значение типа передачи равно 0, означает «синхронный, нециклический», а если значение равно 1-240, означает «синхронный, циклический».

- Только RTR: применимо только к TPDO, которое иницируется полученным сообщением RTR.

Если его значение равно 252 для типа передачи, это означает, что PDO отправляется после получения SYNC и RTR. Значение 253 означает, что PDO отправляется сразу после получения RTR.

- Управляемый событиями: отправляет PDO сразу после возникновения внутреннего события на устройстве CANOpen. Значение 254 указывает пользовательское событие изготовителя устройства. Значение 255 указывает событие, определенное подпротоколом устройства и протоколом прикладного уровня, как правило, изменение значения данных в PDO или времени таймера.

#### Inhibit time

Время отключения определяет минимальный интервал времени между последовательными отправками PDO. Время отключения настроено таким образом, чтобы избежать проблемы слишком частой отправки PDO с высоким приоритетом, всегда занимающей шину, и другие пакеты с более низким приоритетом, которые не могут использовать шину.

#### Event-timer

Используется для указания периодического значения, которое должно быть отправлено. Это 16-разрядное целое число без знака, ед. изм. мс. PDO иницирует отправку с этим значением синхронизации как период. Если значение равно 0, таймер событий не может использоваться.

## Основная функция CANOpen

Характеристика главной станции KS CANOpen:

- Принят стандарт Can2.0A. Соответствует стандарту CANOpen протокола DS301 V4.2.0.
- Поддержка служб управления сетью NMT, включая контроль узлов NMT и контроль ошибок NMT, в качестве главной станции NMT.
- Поддерживается до 32 подчиненных станций CANOpen. Позволяет пользователям настраивать процесс запуска для каждого подчиненного в KincoBuilder.
- Каждая подчиненная станция поддерживает до 8 TPDO и 8 RPDO; Поддерживается до 128 TPDO и 128 RPDO.
- Поддерживает SDO клиент и предоставляет инструкции чтения и записи, которые поддерживают стандартный режим передачи с ускоренным типом.
- Поддерживаются предопределенные экстренные сообщения CANOpen.

## Инструмент управления сетью CANOpen

В KincoBuilder откройте **[Hardware]**, выберите модуль CPU в таблице в верхней части окна, затем нажмите на страницу внизу окна **[CANOpen master station]**, чтобы перейти на страницу конфигурации сети CANOpen.

## EDS файл

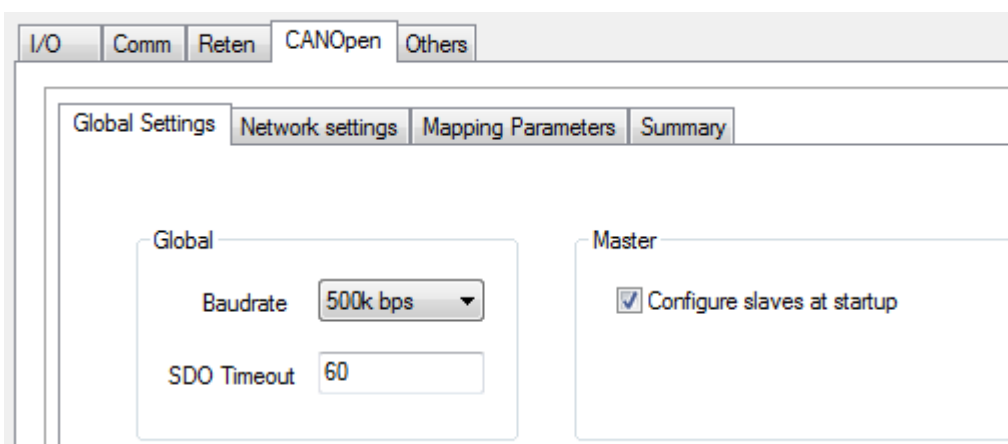
В окне **[CANOpen]** -> **[Setting Network Setting]** для работы с файлами EDS предусмотрены следующие кнопки:

- **[Import EDS]**: Нажмите кнопку, чтобы выбрать файл EDS, импортировать в Kincobuilder и сохранить его. Соответствующее ведомое устройство будет отображаться ниже **[All types of device]**.
- **[Delete]**: Выберите подчиненное устройство из списка всех подчиненных модулей ниже и нажмите кнопку удаления, чтобы удалить устройство из списка, а также его файл EDS из Kincobuilder.
- **[Export All EDS]**: может экспортировать все существующие файлы EDS подчиненного в Kincobuilder в один файл (с именем расширения .ALLEDS). Эта функция полезна при удалении Kincobuilder. Пользователи могут использовать эту функцию для резервного копирования файлов ALLEDS ведомых станций перед удалением, а затем импортировать резервную копию .ALLEDS файлы напрямую.
- **[Import All EDS]**: Файл резервной копии EDS (с расширением .ALLEDS) можно импортировать в Kincobuilder, и все устройства ведомых станций, содержащиеся в этом файле, будут показаны ниже в списке всех модулей ведомых станций.

## Процесс настройки сети CANOpen

1) Настройка глобальных параметров

Перейдите на страницу **[Global Settings]**, как показано ниже:

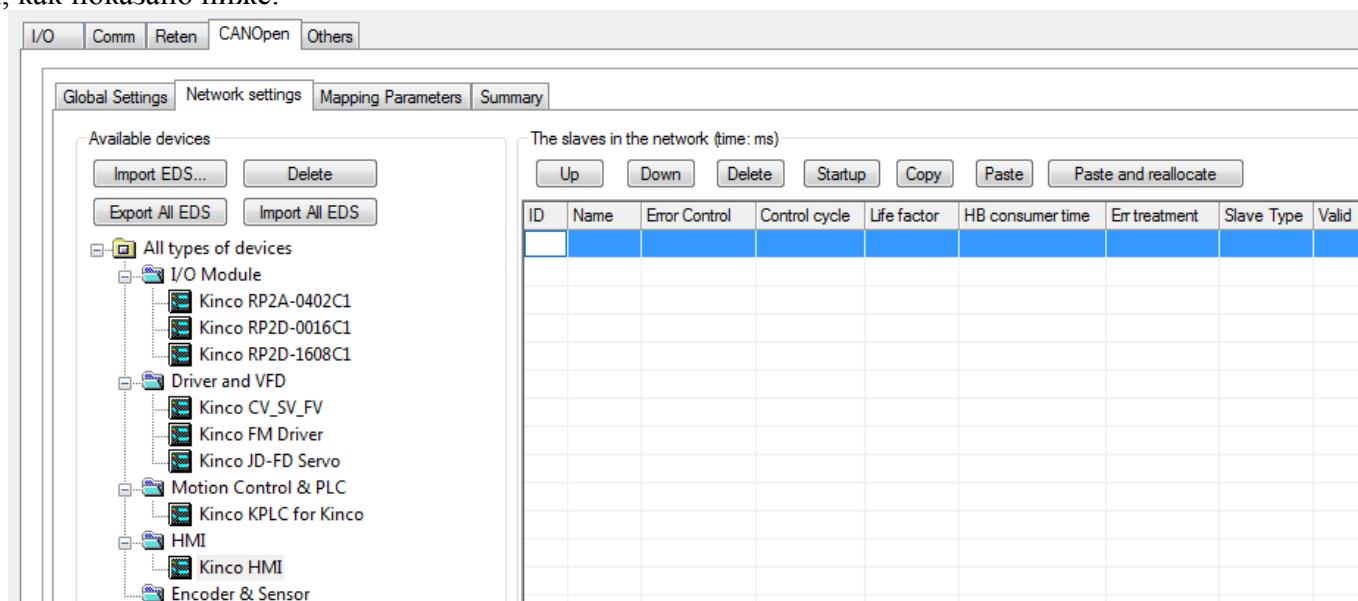


- **[Baudrate]**: выберите скорость передачи, используемую главной станцией. Обратите внимание, что скорость передачи всех узлов в сети должна быть одинаковой.

- **[SDO Timeout]**: установите время ожидания после того, как основная станция отправит сообщение запроса SDO. Если ответное сообщение соответствующего ведомого не получено после этого времени, будет сообщено об ошибке. Значение времени ожидания SDO обычно не должно превышать 100 мс.
- **[Configure slaves at startup]**: если выбрана эта опция, ведущая станция будет не только управлять переходом состояния NMT каждой ведомой станции, но и главная станция будет отправлять соответствующую конфигурацию в соответствии с конфигурацией параметров каждой ведомой станции при запуске. Команда для настройки каждого подчиненного устройства (например, режим контроля ошибок подчиненного устройства, отображение PDO и т. Д.). Если эта опция не выбрана, первичная станция управляет только переходом состояния NMT каждого ведомого.

## 2) Настройка каждой подчиненной станции

Перейдите на страницу **[Network Settings]** и продолжите настраивать подчиненные узлы и их параметры в сети, как показано ниже:



Все функциональные кнопки на странице имеют соответствующие команды меню, вызываемые правой кнопкой мыши. Когда пользователь нажимает правую кнопку мыши в соответствующей позиции, появляется соответствующее меню, вызываемое правой кнопкой мыши, и в это время можно использовать команду меню. Далее описывается общий процесс настройки подчиненного устройства.

### а) Как добавить ведомое устройство в сеть:

Дважды щелкните по ведомому типу, который вы хотите присоединить к сети, в списке слева, добавьте подчиненное устройство этого типа в сеть и оно отобразится в таблице справа.

### б) Как настроить номер станции (ID), тип контроля и другие параметры подчиненного устройства:

**[ID]** в таблице справа - это номер станции ведомого устройства. Первая строка - это местоположение станции 1. Когда вы добавляете новое ведомое устройство, оно отображает параметры конфигурации по умолчанию и Kincobuilder будет добавлять новые устройства в таблицу сверху вниз. Пользователь может щелкнуть строку в таблице, чтобы выбрать подчиненное устройство, а затем нажать кнопки **[Up]** и **[Down]**, чтобы настроить номер его станции или нажать кнопку **[Delete]**, чтобы отключить устройство от сети. **[Error Control]** используется для настройки режима контроля ошибок NMT узла, включая **Node guard** и **Heartbeat**. Если ведомое устройство поддерживает оба режима одновременно, рекомендуется сначала использовать режим **Heartbeat**.

**[Control cycle]** указывает значение периода режима защиты, выбранного ранее. Рекомендуется, чтобы в практических приложениях это значение периода не устанавливалось слишком маленьким, например, оно может быть установлено выше 2000.

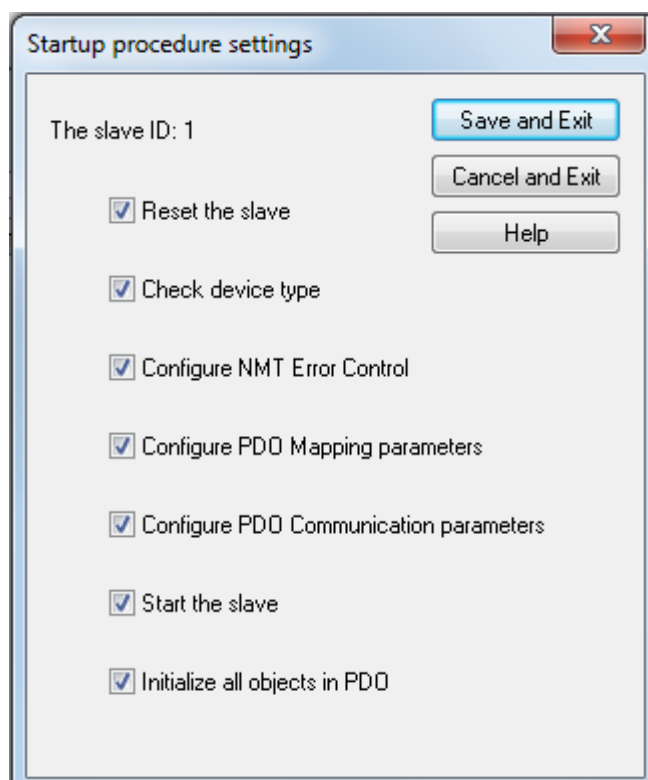
**[HB consumer Time]** Первичная станция будет периодически проверять, получено ли контрольное

сообщение подчиненной станции. Если тайм-аут этого «**heartbeat consumer**» все еще не получен, подчиненная станция считается отключенной, и выполняется соответствующая обработка ошибок. Рекомендуется, чтобы в практических приложениях это значение периода не было слишком маленьким, например, оно может быть установлено выше 2000.

[**Err treatment**] используется для выбора метода обработки, принятого после того, как первичная станция обнаруживает отказ ведомого, включая три параметра: «None», «Stop Node» и «Stop Network». Ошибки, которые может обнаружить первичная станция, включают в себя: тайм-аут команды SDO, нет ответа, защиту узла или тайм-аут сообщения heartbeat, а также прием некоторых типов аварийных сообщений, отправленных вторичной станцией.

### с) Настройка процесса загрузки ведомого:

Выберите ведомое устройство в таблице и нажмите [**Startup**], чтобы выбрать, какую конфигурацию необходимо настроить ведущему устройству для ведомого устройства во время запуска сети.



[**Reset the slave**]: отправка главной станцией команды «Reset Node» перед отправкой команды конфигурации на подчиненную станцию.

[**Check device type**]: перед тем, как главная станция отправит команду конфигурации подчиненной станции, следует ли читать информацию об устройстве для проверки.

[**Configuration NMT error control**]: должна ли ведущая станция настраивать тип контроля ведомой станции и его параметры.

[**Configure PDO Mapping Parameters**]: необходимо ли первичной станции настраивать параметры отображения PDO вторичной станции.

[**Configure PDO communication parameters**]: необходимо ли первичной станцией настраивать параметры связи PDO вторичной станции.

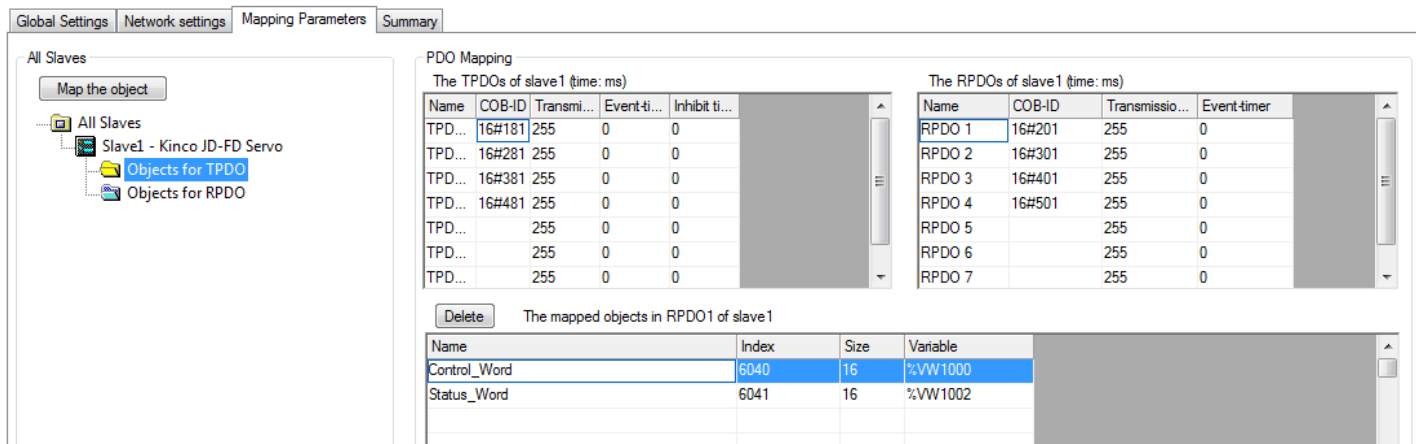
[**Start the slave**]: после завершения настройки главная станция должна отправить команду «Start node» подчиненной станции.

[**Initialize all objects in PDO**]: после запуска подчиненного устройства необходимо ли мастеру очистить данные во всех RPDO подчиненного устройства и немедленно отправить их.



#### д) Настройка PDO каждого ведомого:

Войдите на вкладку **[Mapping parameters]** и настройте отображения PDO ведомых устройств.



В левом столбце **[All Slaves]** перечислены все ведомые устройства в сети и все объекты каждого ведомого, которые могут быть отображены в PDO. **[Objects for TPDO]** могут быть отображены только в TPDO, а **[Objects for RPDO]** могут быть отображены только в RPDO.

Выберите ведомое устройство в списке **[All\_Slaves]**, все PDO этого устройства появятся в таблице с права. Вы можете настраивать каждый PDO.

##### • Параметры связи

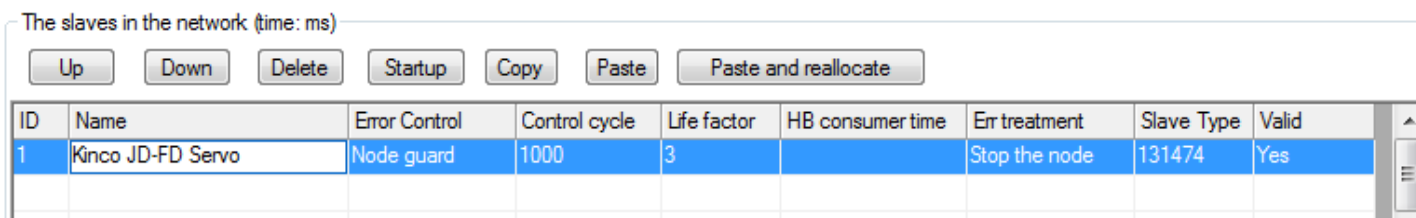
В таблице справа вы можете выбрать PDO для изменения его параметров связи, таких как время синхронизации и время запрета. Среди них нельзя изменять COB-ID первых четырех TPDO и RPDO на подчиненной станции, и используется значение по умолчанию в предопределенном наборе соединений в DS301. Последние четыре TPDO и RPDO позволяют пользователям самостоятельно вводить допустимые значения COB-ID.

##### • Параметры отображения

В списке объектов слева дважды щелкните объект, чтобы добавить его в текущий PDO. В то же время Kincobuilder автоматически назначит этому объекту адрес V-области ПЛК, например VW1006. Пользователь управляет этим адресом V-области в программе. Это эквивалентно работе с соответствующим объектом.

#### е) Копировать / вставить ведомое устройство

- На странице **[Network Setting]** есть 3 кнопки: **[Copy Slave]**, **[Paste Slave]** и **[Paste Slave (Re-divide Memory)]**.



**[Copy Slave]**: выберите настроенное ведомое устройство, а затем нажмите эту кнопку, чтобы скопировать всю информацию ведомого устройства (включая все параметры связи PDO, параметры сопоставления и т. д.). Если выбранное ведомое устройство не сконфигурировано с каким-либо PDO, копирование не выполняется и запрашивается соответствующая информация.

**[Paste Slave]**: после успешного копирования ведомого устройства выберите пустую строку в таблице, а затем нажмите эту кнопку. Информация о ведомом, который вы только что скопировали, будет вставлена в строку, и будет сгенерирован новый ведомый.

**Примечание.** Адрес памяти ПЛК, соответствующий каждому объекту отображения в новом подчиненном PDO, остается таким же, как и в исходном подчиненном устройстве, перераспределения нет, и пользователю необходимо его изменить.

**[Paste and reallocate]:** операция такая же, как и **[Paste Slave]**, но разница состоит в том, что адрес памяти ПЛК каждого объекта отображения в новом подчиненном PDO будет выделяться автоматически без изменения пользователем.

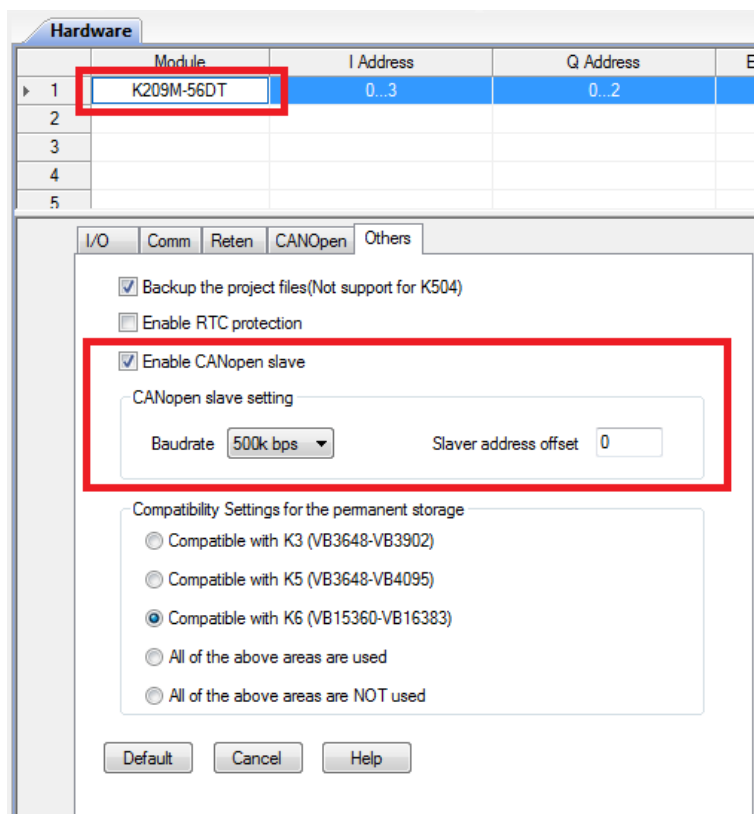
### 4.5.5 Функция CANOpen slave

KS поддерживает функцию ведомого устройства CANOpen и имеет следующие характеристики:

- Интерфейс CAN соответствует стандарту CAN2.0A.
- Совместимость с протоколами DS301 V4.2.0 и DS405 V2.0.
- Поддержка службы управления сетью NMT, поддержка протокола heartbeat.
- Поддержка SDO Server и ускоренный режим передачи.
- Поддерживает до 16 TPDO и 16 RPDO, параметры связи PDO и параметры отображения могут быть свободно настроены.

**Примечание.** При использовании серийный номер PDO должен быть непрерывным, и интервалы не допускаются. Например, можно использовать TPDO1, TPDO2, TPDO3, но нельзя использовать TPDO1, TPDO3, TPDO4.

В пользовательском проекте введите [PLC hardware configuration], а на странице [Other] пользователь может настроить функцию ведомого устройства CANOpen.



**[Enable CANOPEN slave]:** Если выбрана эта опция, этот ПЛК будет действовать как ведомый CANOpen.

**[Baud rate]:** выберите скорость передачи, используемую интерфейсом CAN. Скорость передачи всех узлов в сети должна быть согласованной.

**[Slave Address Offset]:** реальный номер станции этого ведомого устройства равен «комбинированному

значению 1–3-значных DIP-переключателей» плюс «смещение адреса ведомого». Например, если комбинированное значение DIP-переключателей с 1-й по 3-ю цифру равно 5, а смещение адреса подчиненного устройства установлено на 8, номер подчиненной станции равен 13.

#### Object dictionary

Пользователи могут бесплатно загрузить файл .eds этого ведомого устройства с официального сайта Kinco. Файл EDS этого устройства был интегрирован в программное обеспечение KincoBuilder: на странице **[CANOpen Master] → [Network Setting]** конфигурации оборудования, это устройство отображается в **[Motion Control & PLC]** в списке всех подчиненных модулей. Имя подчинённого «**Kinco KPLC for Kinco**». Если вы используете ПЛК Kinco в качестве ведущего и ведомого, вам не нужно повторно импортировать файл eds. Это устройство временно открывает следующие области памяти как объекты связи CANOpen, которые могут быть сопоставлены с PDO для использования. Если вам нужно использовать памяти большего размера, обратитесь в Kinco, чтобы получить другой файл eds.

Индекс объекта	Тип данных	Область памяти ПЛК	Количество	Чтение / запись
0xA040	BYTE	IB0 ~ IB9	10	Только чтение
0xA0C0	INT16	AIW0 ~ AIW18	10	Только чтение
0xA4C0	BYTE	QB0 ~ QB9	10	Чтение и запись
0xA4C1	BYTE	MB0 ~ MB9	10	Чтение и запись
0xA4D2	BYTE	VB508 ~ VB517	10	Чтение и запись
0xA540	INT16	AQW0 ~ AQW18	10	Чтение и запись
0xA550	INT16	VW0 ~ VW18	10	Чтение и запись
0xA641	INT32	VD1016 ~ VD1052	10	Чтение и запись
0xA6C2	FLOAT	VR2032 ~ VR2068	10	Чтение и запись

#### 4.4.6 Функция связи CAN Free

KS предоставляет набор команд связи CAN, которые могут инициализировать порт CAN, отправлять и получать данные через порт CAN и пользователи могут использовать эти команды для связи с другими устройствами. Команда связи CAN поддерживает стандарты CAN2.0A и CAN2.0B. Кроме того, эти команды поддерживают только фреймы данных и не поддерживают удаленные фреймы. Формат кадра данных CAN следующий:

ID	byte 1-8
11 бит (CAN2.0A, стандартный кадр) или 29 бит (CAN2.0B, расширенный кадр)	Данные длиной 1-8 байтов

Функцию связи CANfree можно использовать одновременно с другими функциями связи (шина расширения, Kinco motion control, CANOpen master and slave), но важно отметить, что скорость передачи данных должна быть совместимой.

**Примечание:** ID-номер свободного коммуникационного сообщения не позволяет использовать номер COB-ID в протоколе CANOpen!



## 4.4 Команды CANBus

### 4.5.1 Команды по управлению движением Kinco

Следующие инструкции находятся в группе [Motion Control].

Название	Описание
MC_RPARAS	Чтение параметров привода оси
MC_WPARAS	Изменение параметров привода оси
MC_POWER	Блокировка / разблокировка вала
MC_RESET	Сброс сообщение об ошибке оси и установка в состояние ожидания.
MC_HOME	Поиск нулевой точки
MC_JOG	Функция JOG
MC_MABS	Позиционирование в абсолютной системе
MC_MREL	Позиционирование в относительной системе
MC_MIOT	Чтение номера устройства оси, версии программного обеспечения, ПТ, температуру и т. д.

#### Меры предосторожности

Пользователи должны обратить внимание на следующие моменты при использовании этих инструкций:

- В пользовательском проекте максимально допустимое количество осей: 16 для серий KS и KW и 128 для серии KM.
- Общее количество специальных инструкций, используемых в пользовательском проекте, ограничено: 192 для серий KS и KW и 1024 для серии KM. Среди них, инструкция MC\_MIOT допускает только одну для каждой оси.
- Для той же оси, когда специальная инструкция выполняется и не завершена, нельзя запускать другую специальную инструкцию. Если пользовательская программа запускает другую специальную инструкцию, то инструкция завершится и сообщит об ошибке.
- Для той же оси инструкция MC\_MIOT имеет самый низкий приоритет: если выполняются другие инструкции, инструкция MC\_MIOT не будет выполнена; если выполняется инструкция MC\_MIOT, и программа запускает другие инструкции, инструкция MC\_MIOT прекратит работу.
- Для той же оси, прежде чем пользовательская программа выполнит команду движения (без команды параметра чтения / записи), сначала должна быть выполнена команда MC\_POWER, чтобы удерживать ось. После этого можно продолжить поиск нулевой точки, относительное движение, абсолютное движение или толчковый режим. Если ось не удерживается, то выполнение этих видов инструкций будет завершено и возникнет ошибка.
- Для той же оси пользовательская программа использует команду MC\_RESET для сброса. После успешного сброса ось будет находиться в состоянии ожидания свободной оси. Команда MC\_POWER должна быть выполнена для удержания оси, прежде чем продолжать выполнять перемещение в исходное положение, относительное движение, абсолютное движение или толчковый режим.
- Для команд перемещения в исходное положение, относительного движения, абсолютного движения или толчкового режима ускорение и замедление могут быть заданы командой MC\_WPARAS.
- Вывод каждой команды управления движением, вызываемой в пользовательской программе, не имеет значения. Если инструкция выполнена с ошибкой, ее выходной параметр ERRID выдаст код ошибки, и результат этой ошибки не будет обновлен до следующего повторного выполнения инструкции. Результат выполнения других инструкций не повлияет на результат выполнения инструкции!
- После отключения шины (выход ONLINE команды MC\_STATE равен 1) по соображениям безопасности эта группа инструкций не будет автоматически повторно подключена! После того, как пользователь

исправит ошибку, ПЛК может быть перезапущен после выключения питания!

### Коды ошибок вывода ERRID

Каждая инструкция имеет выходной параметр ERRID. Если инструкция выполнена успешно, вывод ERRID равен 0. Если инструкция не выполняется, для ERRID будет установлено значение кода ошибки, чтобы указать причину ошибки.

Ниже приведено описание каждого значения кода ошибки (обратите внимание, что коды ошибок не подходят для команд MC\_RPARAS и MC\_WPARAS, коды ошибок этих двух инструкций имеют особое значение)

Код ошибки	Описание
0	Без ошибки
1	Ось не активирована или не существует в сети
2	Ось не находится в состоянии удержания.
3	Ось выполняет другие команды управления движением.
4	Буфер передачи сообщений CAN внутри ПЛК заполнен, и сообщения CAN не могут быть отправлены.
5	ПЛК отправил сообщение запроса SDO на ось, но не получил ответ после истечения времени ожидания.
6	ПЛК отправил сообщение запроса SDO на ось, но получил сообщение об ошибке.
7	Инструкция выполняется нормально, но ПЛК постоянно обнаруживает состояние, возвращаемое целевой осью и не определяет правильное значение состояния.

### MC\_RPARAS (чтение параметров) и MC\_WPARAS (изменение параметров)

Цель этой группы инструкций - помочь пользователю управлять параметрами привода в пакетном режиме. Например, пользователь может установить параметры привода на начальном этапе отладки. Пожалуйста, обратитесь к руководству по эксплуатации драйвера для того, чтобы установить конкретные параметры. Если настройки не верны, работа может быть ненормальной. Пожалуйста, действуйте с осторожностью.

#### 1) Список действующих параметров привода

С помощью инструкций чтения и записи привода можно управлять следующими параметрами привода, все параметры доступны для чтения и записи. Каждая инструкция может обрабатывать до 32 параметров одновременно. Тип данных в таблице, REAL означает число с плавающей запятой одинарной точности, UINT32 означает 32-разрядное число без знака, INT32 означает 32-разрядное число со знаком и т.д. Значение «серийный номер» в таблице является фиксированным, и каждый параметр имеет серийный номер. Пользователь может ввести серийный номер в инструкцию для управления соответствующим параметром. «Блок процесса» относится к единице, используемой в параметрах команды, «Диапазон привода» относится к внутреннему диапазону значений привода (эта инструкция автоматически преобразует фактические значения параметров процесса, требуемые пользователем, для внутреннего использования привода. Формат данных, такой же как ускорение, скорость, положение и т.д.).

Серийный номер	Имя параметра	Объект CANOpen	Тип данных	Блок процесса	Диапазон значений
0	Profile_ACC	0x60830020	REAL	Линейное: мм / с <sup>2</sup> Вращение: 1 / с <sup>2</sup>	[0,268435455]
1	Profile_DEC	0x60840020	REAL	Линейное: мм / с <sup>2</sup> Вращение: 1 / с <sup>2</sup>	[0,268435455]

2	Home speed	0x60990120	REAL	Линейное: мм / мин Вращение: град / мин	[-2147483648, 2147483647]
3	Home mode	0x60980008	INT8	DEC	[-128,127]
4	Kvp 0	0x60F90110	UINT16	DEC	[0,32767]
5	Kvi 0	0x60F90210	UINT16	DEC	[0,32767]
6	Kpp 0	0x60FB0110	REAL	HZ	[0,32767]
7	K_Velocity_FF	0x60FB0210	REAL	%	[0,1024]
8	Kvp 1	0x23400410	UINT16	DEC	[0,32767]
9	Kvi 1	0x23400510	UINT16	DEC	[0,32767]
10	Kpp 1	0x23400610	REAL	HZ	[0,32767]
11	CMD_q_Max	0x60730010	UINT16	DEC	[0,2048]
12	Max_speed_RPM	0x607F0020	REAL	Линейное: мм / мин Вращение: град / мин	[-2147483648, 2147483647]
13	Home_Offset_Mode	0x60990508	UINT8	—	[0,255]
14	Motor direction	0x607E0008	UINT8	—	0 и 1
15	Motor_Num2	0x64100110	UINT16	—	[0,65535]
16	Soft_Positive_Limit	0x607D0120	REAL	Линейное: мм Вращение: град	[-2147483648, 2147483647]
17	Soft_Negative_Limit	0x607D0220	REAL	Линейное: мм Вращение: град	[-2147483648, 2147483647]
18	Pos_Filter_N	0x60FB0510	UINT8	—	[0,255]
19	Max_Following_Error	0x60650020	UINT32	DEC	[0,268435455]
20	Target_Pos_Window	0x60670020	UINT32	DEC	[0,268435455]
21	Position_Window_time	0x60680010	UINT16	DEC	[0,32767]
22	Speed_Fb_N	0x60F90508	REAL	HZ	[0,45]
23	Speed_Mode	0x60F90608	UINT8	—	[0,85]
24	Input port polarity	0x20100110	UINT8	—	[0,255]
25	DIN1	0x20100310	UINT16	—	[0,65535]
26	DIN2	0x20100410	UINT16	—	[0,65535]
27	DIN3	0x20100510	UINT16	—	[0,65535]
28	DIN4	0x20100610	UINT16	—	[0,65535]
29	DIN5	0x20100710	UINT16	—	[0,65535]
30	DIN6	0x20100810	UINT16	—	[0,65535]
31	DIN7	0x20100910	UINT16	—	[0,65535]
32	DIN8	0x20101D10	UINT16	—	[0,65535]
33	Output port polarity	0x20100D10	UINT8	—	[0,255]
34	OUT1	0x20100F10	UINT16	—	[0,65535]

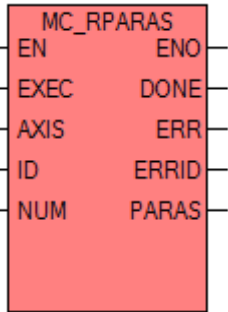
35	OUT2	0x20101010	UINT16	—	[0,65535]
36	OUT3	0x20101110	UINT16	—	[0,65535]
37	OUT4	0x20101210	UINT16	—	[0,65535]
38	OUT5	0x20101310	UINT16	—	[0,65535]
39	OUT6	0x20101E10	UINT16	—	[0,65535]
40	OUT7	0x20101F10	UINT16	—	[0,65535]
41	OUT8	0x25080420	UINT32	DEC	[-2147483648, 2147483647]
42	Pulse mode	0x25080308	UINT8	—	[0,255]
43	Save control parameter	0x10100120	UINT32	—	Только 16 # 65766173
44	Init control parameters	0x10110120	UINT32	—	Только 16 # 65766173

## 2) Описание параметра ERRID

Инструкции для параметров чтения и записи предоставляют выходные параметры ERRID (тип DWORD). Это значение параметра является кодом ошибки, указывающим на ошибку, которая произошла во время выполнения инструкции.

Код ошибки	Описание
0xFFFFFFFF	Произошла ошибка, которая вызвала сбой: 1) Неверный номер оси, введенный пользователем и количество параметров. 2) Есть другие специальные инструкции Kinco. 3) Инструкция должна работать с 32 параметрами.
Другое значение	Каждый бит ERRID указывает результат операции соответствующего параметра и каждый бит соответствует одному параметру, указанному в таблице номеров параметров идентификатора: бит 0 указывает результат первого параметра текущей операции, а бит 1 обозначает результат второго параметра операции и тд. Значение бита равное 1 указывает на то, что соответствующая операция параметра не выполнена, в противном случае соответствующая операция параметра завершается успешно.

## MC\_RPARAS (чтение параметра)

	Название	Применение	Подходит для
LD	MC_RPARAS		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
<i>EXEC</i>	Вход	BOOL	V, M, L, SM	Инструкция выполняется по переднему фронту EXEC.
<i>AXIS</i>	Вход	INT	V, M, L, константа	Номер оси (то есть адрес ведомого CANOpen)
<i>ID</i>	Вход	BYTE	V, M, L	Начальный адрес таблицы порядкового номера считываемого параметра.
<i>NUM</i>	Вход	INT	V, M, L, константа	Количество параметров для чтения
<i>DONE</i>	Выход	BOOL	V, M, L	Когда выполнение инструкции завершено, DONE переходит от 0 к 1.
<i>ERR</i>	Выход	BOOL	V, M, L	Равен 1, если при выполнении инструкции произошла ошибка.
<i>ERRID</i>	Выход	DWORD	V, M, L	Код ошибки
<i>PARAS</i>	Выход	DWORD	V, M, L	Начальный адрес хранения всех значений параметров.

**Примечание:** AXIS и NUM должны быть либо все константы, либо все переменные. Кроме того, параметры ID и PARAS вместе образуют блок памяти переменной длины. Все блоки памяти должны находиться в допустимой области памяти, в противном случае результат будет непредсказуем.

Три параметра ID, PARAS и NUM вместе образуют таблицу параметров. ID является начальным адресом таблицы порядковых номеров. Серийный номер каждого параметра, который должен работать, сохраняется последовательно от этого адреса (то есть «серийный номер» в предыдущем списке параметров), каждый порядковый номер занимает 1 байт; PARAS - это значение параметра. Начальный адрес таблицы, из которой последовательно хранятся значения параметров чтения, каждый из которых занимает 4 байта; NUM - это количество параметров, которые будут использоваться. Например, предположим, что параметром ID является VB100, параметром PARAS является VD1200, а параметром NUM - 3. Затем VB100, VB101 и VB102 соответственно сохраняют серийные номера трех параметров, которые должны работать, и три параметра, считанные после завершения выполнения инструкции. Значения хранятся в VD1200, VD1204 и VD1208.

Обратите внимание, что для PARAS, хотя таблица значений параметров использует адрес DWORD равномерно, фактические типы данных различных параметров процесса не совпадают. Следовательно, в пользовательской программе пользователь должен обрабатывать данные в таблице параметров в соответствии с фактическим типом данных.

- Если фактический тип данных процесса имеет тип REAL, то память параметров может напрямую управляться адресом с плавающей запятой. Например, если значение параметра хранится в VD1200, тогда VR1200 может работать напрямую. Потому что VD1200 и VR1200 фактически занимают один и тот же адрес памяти в ПЛК.
- Если фактический тип данных процесса отличается от REAL, и соответствующая память параметров не вынуждает определять тип данных в таблице глобальных переменных, то память параметров может быть считана напрямую, потому что инструкция будет автоматически обрабатывать различные целые числа с знаком и без знака. Например, если значение параметра хранится в VD1200, а фактическим типом является INT32 или UINT32, то VD1200 работает напрямую.

Описание инструкции в формате LD:

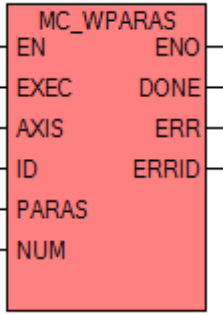
Если EN равен 1, то команда запускается для выполнения по переднему фронту на входе EXEC. Эта инструкция последовательно отправляет SDO драйверу для считывания соответствующего объекта в

соответствии с таблицей параметров, которая должна быть считана с помощью идентификатора ID и NUM, и будет считывать соответствующий объект. Данные помещаются в таблицу значений, указанную PARAS по очереди, и соответствующий бит ERRID устанавливается в 0. Если ответ SDO параметра неверен или тайм-аут не отвечает, данные соответствующего адреса в PARAS остаются без изменений, и соответствующий бит ERRID устанавливается в 1, а затем читается следующий параметр. Когда все параметры считаны, DONE устанавливается в 1, а ERR и ERRID устанавливаются в разные значения в соответствии с результатом выполнения.

Если EN равен 0, инструкция не выполняется. Когда EXEC становится 0 во время выполнения инструкции, инструкция прекращает чтение параметров, которые не были завершены, и устанавливает DONE в 1, ERR и ERRID сохраняют выполненные результаты.

Если ПЛК обнаружит ошибку, при запуске команды (например, ось не активирована, ось выполняет другие команды и т.д.), DONE и ERR установятся в 1 и ERRID покажет соответствующий код ошибки.

### MC\_WPARAS (изменение параметров)

	Название	Применение	Подходит для
LD	MC_WPARAS		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
<i>EXEC</i>	Вход	BOOL	V, M, L, SM	Инструкция выполняется по переднему фронту EXEC.
<i>AXIS</i>	Вход	INT	V, M, L, константа	Номер оси (то есть адрес ведомого CANOpen)
<i>ID</i>	Вход	BYTE	V, M, L	Начальный адрес таблицы порядкового номера изменяемого параметра.
<i>PARAS</i>	Вход	DWORD	V, M, L	Чтение начального адреса сохранённых значений параметров.
<i>NUM</i>	Вход	INT	V, M, L, константа	Количество параметров для записи
<i>DONE</i>	Выход	BOOL	V, M, L	Когда выполнение инструкции завершено, DONE переходит от 0 к 1.
<i>ERR</i>	Выход	BOOL	V, M, L	Равен 1, если при выполнении инструкции произошла ошибка.
<i>ERRID</i>	Выход	DWORD	V, M, L	Код ошибки

**Примечание:** AXIS и NUM должны быть либо все константы, либо все переменные. Кроме того, параметры ID и PARAS вместе образуют блок памяти переменной длины. Все блоки памяти должны находиться в



допустимой области памяти, в противном случае результат будет непредсказуем.

Три параметра ID, PARAS и NUM вместе образуют таблицу параметров. ID является начальным адресом таблицы порядковых номеров. Серийный номер каждого параметра, который должен работать, сохраняется последовательно от этого адреса (то есть «серийный номер» в предыдущем списке параметров), каждый порядковый номер занимает 1 байт; PARAS - это значение параметра. Начальный адрес таблицы, из которой последовательно хранятся значения параметров чтения, каждый из которых занимает 4 байта; NUM - это количество параметров, которые будут использоваться. Например, предположим, что параметром ID является VB100, параметром PARAS является VD1200, а параметром NUM - 3. Затем VB100, VB101 и VB102 соответственно сохраняют серийные номера трех параметров для работы, и VD1200, VD1204 и VD1208 соответственно сохраняют параметры, подлежащие изменению.

**Примечание:** для PARAS, хотя таблица значений параметров использует адрес DWORD, фактический тип данных каждого параметра процесса не совпадает, поэтому в пользовательской программе в таблице пользователь должен присвоить значение соответствующему адресу в таблице параметров в соответствии с фактическим типом данных.

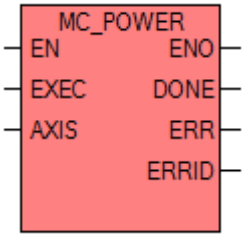
- Если фактический тип данных процесса имеет тип REAL, то память параметров может управляться напрямую по адресу с плавающей запятой. Например, если предполагается, что значение параметра будет сохранено в VD1200, тогда VR1200 может работать напрямую. VD1200 и VR1200 фактически занимают один и тот же адрес памяти в ПЛК и инструкция автоматически выполнит преобразование типов.
- Если фактический тип данных процесса отличается от REAL, памятью параметров можно напрямую манипулировать и инструкция автоматически выполнит преобразование типа в соответствии с типом данных параметра. Например, если тип данных параметра - UINT16, то VD1200 может быть напрямую назначено допустимое значение.

### Описание инструкции в формате LD

Если EN равен 1, то команда запускается для выполнения по переднему фронту на входе EXEC. Инструкция последовательно отправляет значение в PARAS на накопитель через SDO, чтобы изменить соответствующий объект в соответствии с таблицей параметров, указанной в ID, PARAS, NUM. Соответствующий бит ERRID установлен на 0. Если SDO параметра отвечает неправильно или тайм-аут не отвечает, соответствующий бит ERRID будет установлен на 1, а затем продолжит записывать следующий параметр. Когда все параметры записаны, для DONE устанавливается значение 1, а для ERR и ERRID устанавливаются разные значения в соответствии с результатом выполнения.

Если EN равен 0, инструкция не выполняется. Если EN становится 0 во время выполнения инструкции, инструкция прекращает запись параметров, которые не были завершены, и устанавливает DONE в 1, ERR и ERRID поддерживают выполненный результат. Если ПЛК обнаружит ошибку при запуске команды (например, ось не активирована, ось выполняет другие команды и т.д.), DONE и ERR установятся в 1, а в ERRID соответствующий код ошибки.

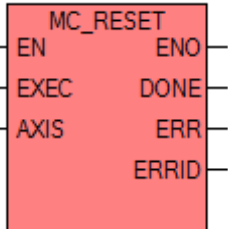
## MC\_POWER (блокировка и ослабление вала)

	Название	Применение	Подходит для
LD	MC_POWER		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
<i>EXEC</i>	Вход	BOOL	V, M, L, SM	Передний фронт запускает команду блокировки оси, а задний фронт запускает команду ослабления оси.
<i>AXIS</i>	Вход	INT	V, M, L, константа	Номер оси (то есть адрес ведомого CANOpen)
<i>DONE</i>	Выход	BOOL	V, M, L	Когда выполнение инструкции завершено, DONE переходит от 0 к 1.
<i>ERR</i>	Выход	BOOL	V, M, L	Равен 1, если при выполнении инструкции произошла ошибка.
<i>ERRID</i>	Выход	BYTE	V, M, L	Код ошибки

Если EN равен 1, то выполнение команды блокировки оси будет запускаться по переднему фронту EXEC, а выполнение команды освобождения оси будет запускаться по заднему фронту EXEC. Когда инструкция выполняется, ПЛК сначала отправляет команду для управления осью, чтобы войти в состояние ожидания и проверяет фактическое состояние возврата привода в течение периода ожидания 5 секунд. Если выполнение прошло успешно, DONE установлен на 1, ERR установлен на 0 и ERRID установлен на 0. Если происходит ошибка (может быть выполнение самой инструкции или драйвер не выполнил действие правильно во время выполнения, см. код ошибки), инструкция не будет выполнена, инструкция прекратит выполнение, и DONE будет установлен на 1, ERR и ERRID присваивается соответствующий код ошибки. Если EN равен 0, инструкция не выполняется.

### MC\_RESET (сброс аварийного сигнала привода)

	Название	Применение	Подходит для
LD	MC_RESET		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
<i>EXEC</i>	Вход	BOOL	V, M, L, SM	Команда выполняется по переднему фронту один раз.
<i>AXIS</i>	Вход	INT	V, M, L, константа	Номер оси (то есть адрес ведомого CANOpen)
<i>DONE</i>	Выход	BOOL	V, M, L	Когда выполнение инструкции завершено, DONE переходит от 0 к 1.
<i>ERR</i>	Выход	BOOL	V, M, L	Равен 1, если при выполнении инструкции произошла ошибка.
<i>ERRID</i>	Выход	BYTE	V, M, L	Код ошибки

Когда ось совершает ошибку во время работы, эта инструкция может быть вызвана для сброса информации

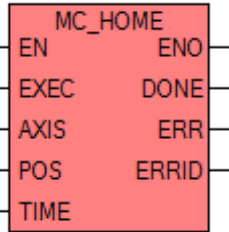


об ошибке на оси и в то же время для перевода в состояние ожидания свободного вала. Если вам необходимо продолжить выполнение других команд движения после успешного сброса, вы сначала должны вызвать инструкцию MC\_POWER, чтобы заблокировать ось.

**Примечание:** эта инструкция сбрасывает только информацию об ошибке драйвера, но не сбрасывает результаты вывода каждой инструкции.

Если EN равен 1, выполнение этой инструкции будет запущено по переднему фронту EXEC. Когда инструкция выполняется, ПЛК сначала отправляет команду для сброса аварийного сигнала привода и проверяет фактическое состояние привода в течение периода ожидания 2 секунды. Если сброс выполнен успешно, это означает, что команда была успешно выполнена, DONE установлен на 1, ERR установлен на 0 и ERRID установлен на 0. Если возникает ошибка (это может быть ошибка в выполнении самой инструкции или ошибка, что привод не выполнил действие правильно, см. код ошибки), выполнение инструкции завершается неудачно, инструкция прекращает выполнение и DONE устанавливается в 1, ERR устанавливается в 1, а ERRID отображает соответствующий код ошибки. Если EN равен 0, инструкция не выполняется.

## MC\_HOME (Поиск нулевой точки)

	Название	Применение	Подходит для
LD	MC_HOME		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
EXEC	Вход	BOOL	V, M, L, SM	Передний фронт запускает эту команду один раз; задний фронт останавливает движение
AXIS	Вход	INT	V, M, L, константа	Номер оси (то есть адрес ведомого CANOpen)
POS	Вход	REAL	V, M, L, константа	Исх. смещение, ед. изм.: мм или град.
TIME	Вход	DWORD	V, M, L, константа	Если точка не найдена в течение этого времени, команда завершится с ошибкой.
DONE	Выход	BOOL	V, M, L	Когда выполнение инструкции завершено, DONE переходит от 0 к 1.
ERR	Выход	BOOL	V, M, L	Равен 1, если при выполнении инструкции произошла ошибка.
ERRID	Выход	BYTE	V, M, L	Код ошибки

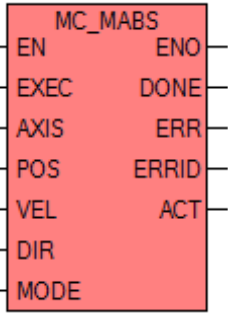
Выполнив эту команду, ось может быть возвращена в начало координат. Параметр POS устанавливает значение смещения исходных координат.

**Примечание:** В этой инструкции используется режим внутреннего наведения драйвера. Сначала вы должны установить режим наведения 60980008 на драйвер (также записанный инструкцией MC\_WPARAS). Для

получения подробной информации, пожалуйста, обратитесь к руководству по эксплуатации. Если EN равен 1, выполнение этой инструкции будет запущено по переднему фронту EXEC. Когда инструкция выполняется, ПЛК сначала отправляет команду запустить ось, чтобы найти источник; после завершения передачи проверьте статус возврата драйвера. Проверка будет продолжаться в течение времени TIME (время ожидания, установленное пользователем, единица измерения мс). Если ось успешно находит источник в течение этого времени, это означает, что инструкция была выполнена успешно. В это время для DONE установлено значение 1, для ERR установлено значение 0, а для ERRID установлено значение 0. Если возникает ошибка (это может быть ошибка при выполнении самой инструкции или ошибка во время выполнения драйвера, см. код ошибки), выполнение инструкции завершается неудачно, инструкция прекращает выполнение, и DONE устанавливается на 1, а ERR устанавливается на 1, ERRID отображает соответствующий код ошибки. Если EN равен 0, инструкция не выполняется.

Если EN становится равным 0 во время выполнения, инструкция прекращает выполнение и ось будет находиться в состоянии блокировки и ожидания.

## MC\_MABS (PABS)

	Название	Применение	Подходит для
LD	MC_MABS		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
EXEC	Вход	BOOL	V, M, L, SM	Передний фронт запускает эту команду один раз; задний фронт вызывает паузу в движении.
AXIS	Вход	INT	V, M, L, константа	Номер оси (то есть адрес ведомого CANOpen)
POS	Вход	REAL	V, M, L, константа	Абсолютное положение, ед. изм.: мм или °
VEL	Вход	REAL	V, M, L, константа	Максимальная скорость (> 0) увеличивается во время движения, ед. изм.: мм / мин или ° / мин.
DIR	Вход	INT	V, M, L, константа	Направление движения. Зарезервировано. Этот параметр может быть установлен в 0.
MODE	Вход	INT	V, M, L, константа	Режим движения: 0 = одиночное выполнение, команда завершится после того, как ось выполнит абсолютное позиционирование. 1 = постоянное выполнение. После того, как ось выполнит абсолютное позиционирование, команда не завершится. Если найдена новая целевая позиция, будет отправлена команда, позволяющая оси продолжить выполнять новое абсолютное

				позиционирование.
<i>DONE</i>	Выход	BOOL	V, M, L	Когда выполнение инструкции завершено, DONE переходит от 0 к 1.
<i>ERR</i>	Выход	BOOL	V, M, L	Равен 1, если при выполнении инструкции произошла ошибка.
<i>ERRID</i>	Выход	BYTE	V, M, L	Код ошибки
<i>ACT</i>	Выход	BOOL	V, M, L	<b>MODE = 0</b> , одиночное выполнение, ACT указывает, правильно ли активирована одиночная инструкция позиционирования. 1 означает активный, 0 означает неактивный. <b>MODE = 1</b> , когда выполняется постоянно, ACT указывает, правильно ли активирована инструкция постоянного позиционирования. 1 означает активный (он останется равным 1 после завершения одиночного позиционирования), 0 означает, что не активен.

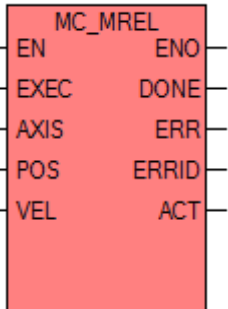
Эта команда управляет осью для перемещения в абсолютной системе. При движении скорость начинается с текущего значения и достигает нуля, когда достигает позиции. Эта команда позволяет сделать паузу.

Если EN равен 1, выполнение этой инструкции будет запущено по переднему фронту EXEC.

Когда инструкция выполняется, ПЛК управляет осью, чтобы начать абсолютное позиционирование оси в соответствии со значениями параметров позиции (POS) и скорости движения (VEL), введенными пользователем. Во время движения команда будет непрерывно сканировать позицию и значение параметра скорости. Если есть изменения, команда будет немедленно отправлена на ось, то есть новые параметры скорости и значения параметров положения могут быть приняты в любое время (например, для выполнения паузы во время движения установите скорость на 0). В то же время ПЛК будет постоянно проверять состояние возврата оси. Если позиция достигнута успешно, DONE установится на 1, ERR установится на 0 и ERRID установится на 0. После завершения позиционирования команда оценит значение MODE, если он установлен в режим одиночной работы, команда завершится; если он установлен в постоянный режим работы, команда не будет завершена, и значение положения будет сканироваться всё время. Когда POS изменится, ось выполнит новое абсолютное позиционирование.

Если возникает ошибка (это может быть ошибка при выполнении самой инструкции или ошибка привода, подробнее см. код ошибки), выполнение инструкции завершится неудачно, инструкция прекращает выполнение и DONE установится на 1, ERR установится на 1, ERRID присвоит соответствующий код ошибки. Если EN равен 0, инструкция не выполняется. Если EN становится равным 0 во время выполнения, инструкция прекратит выполнение и ось будет находиться в состоянии блокировки и ожидания.

## MC\_MREL (PREL)

	Название	Применение	Подходит для
LD	MC_MREL		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
<i>EXEC</i>	Вход	BOOL	V, M, L, SM	Передний фронт запускает эту команду один раз; задний фронт вызывает паузу в движении.
<i>AXIS</i>	Вход	INT	V, M, L, константа	Номер оси (то есть адрес ведомого CANOpen)
<i>POS</i>	Вход	REAL	V, M, L, константа	Относительное расстояние перемещения в мм или °. Положительное число указывает на движение в положительном направлении; отрицательное число указывает движение в отрицательном направлении.
<i>VEL</i>	Вход	REAL	V, M, L, константа	Максимальная скорость (> 0) увеличивается во время движения, ед. изм.: мм / мин или ° / мин.
<i>DONE</i>	Выход	BOOL	V, M, L	Когда выполнение инструкции завершено, DONE переходит от 0 к 1.
<i>ERR</i>	Выход	BOOL	V, M, L	Равен 1, если при выполнении инструкции произошла ошибка.
<i>ERRID</i>	Выход	BYTE	V, M, L	Код ошибки
<i>ACT</i>	Выход	BOOL	V, M, L	Показывает, правильно ли активирована команда. 1 означает активный, 0 означает неактивный.

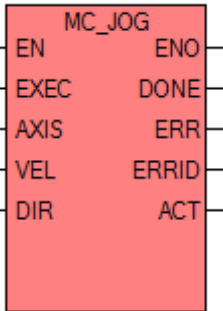
Эта команда управляет осью для перемещения на указанное расстояние POS (используя текущую позицию в качестве начальной позиции). При движении скорость начинается с текущего значения и достигает нуля, когда ось достигает заданной позиции. Эта инструкция позволяет сделать паузу.

Если EN равен 1, выполнение этой инструкции будет запущено по переднему фронту EXEC.

Когда инструкция выполняется, ПЛК управляет осью, чтобы начать относительное позиционирование в соответствии со значениями параметров (POS) и скорости движения (VEL), введенными пользователем. Во время движения команда будет непрерывно сканировать значение параметра скорости. Если есть изменение, команда будет немедленно отправлена на ось, то есть новое значение параметра скорости может быть принято в любое время (например, для выполнения паузы установите скорость на 0 во время движения, а затем возобновите движение, снова указав значение скорости). В то же время ПЛК будет постоянно проверять состояние оси. Если позиция достигнута успешно, DONE установится на 1, ERR установится на 0, а ERRID установится на 0. Если происходит ошибка (это может быть ошибка в выполнении самой

инструкции или ошибка привода, подробности см. в коде ошибки), выполнение инструкции прекращается и DONE установится в 1, ERR установится в 1, ERRID присвоит соответствующий код ошибки. Если EN равен 0, инструкция не выполняется. Если EN становится равным 0 во время выполнения, инструкция прекращает выполнение и ось будет находиться в состоянии блокировки и ожидания.

## MC\_JOG (JOG)

	Название	Применение	Подходит для
LD	MC_JOG		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
<i>EXEC</i>	Вход	BOOL	V, M, L, SM	Передний фронт запускает эту команду один раз; задний фронт вызывает паузу в движении.
<i>AXIS</i>	Вход	INT	V, M, L, константа	Номер оси (то есть адрес ведомого CANOpen)
<i>VEL</i>	Вход	REAL	V, M, L, константа	Скорость движения, ед. : мм / мин или ° / мин. Положительное число указывает на положительное направление, а отрицательное число указывает на отрицательное направление.
<i>DIR</i>	Вход	INT	V, M, L, константа	Направление движения. Зарезервировано. Этот параметр может быть установлен в 0.
<i>DONE</i>	Выход	BOOL	V, M, L	Когда выполнение инструкции завершено, DONE переходит от 0 к 1.
<i>ERR</i>	Выход	BOOL	V, M, L	Равен 1, если при выполнении инструкции произошла ошибка.
<i>ERRID</i>	Выход	BYTE	V, M, L	Код ошибки
<i>ACT</i>	Выход	BOOL	V, M, L	Показывает, правильно ли активирована команда. 1 означает активный, 0 означает неактивный.

Эта команда контролирует ось для движения с целевой скоростью, указанной в VEL.

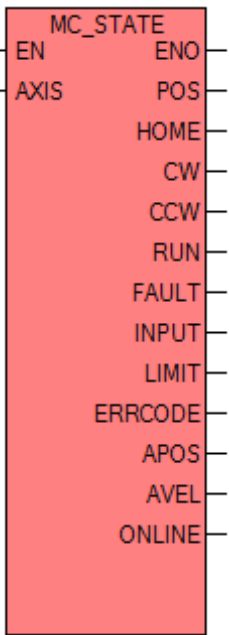
Если EN равен 1, выполнение этой инструкции будет запущено по переднему фронту EXEC.

Когда инструкция выполняется, ПЛК управляет осью, чтобы начать толчковый режим в соответствии с введенным пользователем значением скорости (VEL). Во время движения оси команда будет непрерывно сканировать значение параметра скорости. Если есть изменение, команда будет немедленно отправлена на ось, то есть новые значения параметров скорости могут быть приняты в любое время.

Если возникает ошибка (это может быть ошибка при выполнении самой инструкции или ошибка привода, подробнее см. код ошибки), инструкция прекращает выполнение и DONE установится на 1, ERR

установится на 1, ERRID присвоит соответствующий код ошибки. Если EN равен 0, инструкция не выполняется. Если EXEC становится равным 0 во время выполнения, инструкция прекратит выполнение и ось перейдет в состоянии блокировки и ожидания.

### MC\_STATE (чтение состояние привода)

	Название	Применение	Подходит для
LD	MC_STATE		KS105C1 KS105C2

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
<i>AXIS</i>	Вход	INT	V, M, L, константа	Номер оси (то есть адрес ведомого CANOpen)
<i>POS</i>	Выход	BOOL	V, M, L	Сигнал "Позиция достигнута"
<i>HOME</i>	Выход	BOOL	V, M, L	Сигнал "Нулевая точка"
<i>CW</i>	Выход	BOOL	V, M, L	Сигнал "Двигатель вперед"
<i>CCW</i>	Выход	BOOL	V, M, L	Сигнал "Двигатель назад"
<i>RUN</i>	Выход	BOOL	V, M, L	Сигнал "Двигатель работает"
<i>FAULT</i>	Выход	BOOL	V, M, L	Сигнал "Ошибка оси"
<i>INPUT</i>	Выход	WORD	V, M, L	Состояние DI, BIT0 ~ BIT7 соответствует состоянию DNI1 ~ DIN8.
<i>LIMIT</i>	Выход	BOOL	V, M, L	Сигнал «Достигнут предел»
<i>ERRCODE</i>	Выход	WORD	V, M, L	Код ошибки
<i>APOS</i>	Выход	REAL	V, M, L	Текущее фактическое положение машины, единица измерения в мм или °
<i>AVEL</i>	Выход	REAL	V, M, L	Фактическая скорость машины, единица измерения в мм / мин или ° / мин.



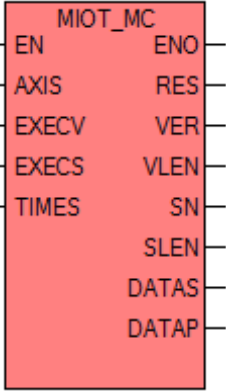
ONLINE	Выход	BYTE	V, M, L	Сигнал состояния оси. 1 означает, что ось отключена, 0 означает, что ось подключена.
--------	-------	------	---------	--

Эта инструкция постоянно сканирует состояние драйвера, получает флаги различных состояний и выводит их в соответствующие выходные параметры.

**Примечание:** Два сигнала «Позиция достигнута» и «Нулевая точка» изменяют состояние на 0 во время выполнения действия (позиционирование или поиск нулевой точки) и не будут установлены в 1, пока действие не будет выполнено правильно!

Если EN равен 1, эта инструкция выполняется. Если EN равен 0, инструкция не выполняется и выходные параметры не обновляются.

## MIOT\_MC (чтение информации о драйвере сервопривода Kinco)

	Название	Применение	Подходит для
LD	MC_MIOT		KS105C1 KS105C2

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
AXIS	Вход	INT	V, M, L, константа	Номер оси (адрес ведомого CANOpen)
EXECV	Вход	BOOL	V, M, L, SM	Чтение серийного номера и версии программного обеспечения по переднему фронту.
EXECS	Вход	BOOL	V, M, L, SM	Чтение ПТ, температуры, времени работы по переднему фронту.
TIMES	Вход	INT	V, M, L	Таймер, чтения ИИТ, температуры, время работы. Если = 0, таймер не включается.
RES	Выход	BYTE	V, M, L	Результат
VER	Выход	BYTE	V, M, L	Информация о версии программного обеспечения
VLEN	Выход	BYTE	V, M, L	Общая длина информации о версии программного обеспечения, в байтах
SN	Выход	BYTE	V, M, L	Информация о серийном номере
SLEN	Выход	BYTE	V, M, L	Общая длина информации о серийном номере, в байтах
DATAS	Выход	BYTE	V, M, L	Начальный адрес ПТ, температуры, время работы

DATA#	Выход	BYTE	V, M, L	Слово состояния, ток, скорость и другая информация, хранящаяся по начальному адресу
-------	-------	------	---------	---

Эта инструкция используется для считывания рабочего состояния и другой информации целевой оси. Только одна из этих инструкций допускается для каждой оси.

Для той же оси инструкция MC\_MIoT имеет самый низкий приоритет: если выполняется другая инструкция движения, MC\_MIoT не будет выполняться; Если запускаются другие инструкции движения, инструкция MC\_MIoT прерывается и завершается.

Информация, прочитанная в этой инструкции, разделена на три категории, как описано ниже.

#### • Серийный номер и версия программного обеспечения

Эта информация будет прочтена по переднему фронту параметра EXECV, читается один раз. Это фиксированная информация, как правило, может быть прочтена один раз при включении питания.

Параметр VER указывает начальный адрес для информации о версии программного обеспечения, которая постоянно сохраняется в той области. Значение VLEN указывает общую длину информации о версии программного обеспечения, то есть количество использованных байтов.

Параметр SN указывает начальное местоположение для информации о серийном номере продукта, которая постоянно хранится в начальном адресе. Значение SLEN указывает общую длину информации о серийном номере, то есть количество использованных байтов.

После каждого запуска, ПЛК выполняет процесс чтения. Если все успешно прочитано, данные и длина в выходных параметрах обновляются. Если чтение не удалось, выходные параметры не обновляются.

В случае успеха или неудачи соответствующий бит в RES сбрасывается после выполнения инструкции.

#### • ПТ, температура привода, информация о времени работы

Это информация, которую нужно читать в режиме реального времени, но не так часто, чтобы она могла мешать другим инструкциям.

Существует два вида условий запуска для чтения этих параметров: передний фронт параметра EXECS запускает чтение один раз; цикл считывания времени, указанный в TIMES, при котором ПЛК будет запускать чтение каждый цикл. Если значение параметра TIMES равно 0, чтение по времени не будет работать.

Параметр DATAS указывает начальный адрес для хранения этой информации и информация о каждом параметре сохраняется, как показано в таблице ниже:

Параметр	Адрес	Тип данных	Длина	Смещение в байтах
ПТ	0x60F612	UINT16	2	0
Температура	0x60F70B	UINT16	2	2
Время работы	0x2FF700	UINT32	4	4

После каждого запуска ПЛК выполняет процесс чтения. Если все успешно прочитано, данные и длина в выходных параметрах обновляются. Если чтение не удалось, выходные параметры не обновляются.

В случае успеха или неудачи соответствующий бит в RES сбрасывается после выполнения инструкции.

#### • Слово состояния, слово ошибки, фактический ток и т.д.

Эта информация автоматически читается инструкциями через PDO, без необходимости запуска пользователем программы.

Параметры АТАР указывают начальный адрес, где будет храниться информация и информация о каждом параметре будет сохраняться, как показано в таблице ниже:

Параметр	Адрес	Тип данных	Длина	Смещение в байтах
Слово состояния	0x604100	UINT16	2	0



Слово ошибки	0x260100	UINT16	2	2
Слово ошибки 1	0x260200	UINT16	2	4
Фактический ток	0x607800	INT16	2	6
Фактическая скорость	0x606C00	INT32	4	8
Фактическая позиция	0x606300	INT32	4	12

#### ● Результат выполнения RES

Бит в RES	Описание
Бит 7	Указывает, завершено ли чтение серийного номера и версии программного обеспечения. 0 означает, что он находится в процессе чтения; 1 означает, что чтение завершено (успешно или неудачно)
Бит 6	Указывает, произошла ли ошибка во время чтения серийного номера и версии программного обеспечения. 0 означает успешное чтение; 1 указывает на ошибку при чтении.
Бит 5	Указывает, завершено ли чтение ПТ, температуры привода и времени работы. 0 означает, что он находится в процессе чтения; 1 означает, что чтение завершено (успешно или неудачно)
Бит 4	Указывает, произошла ли ошибка во время чтения ПТ, температуры привода и времени работы. 0 означает успешное чтение; 1 указывает на ошибку при чтении.
Бит 3...0	Объединенное значение представляет ошибку выполнения: 0 --- нет ошибок 1 --- номер оси неверен 2 --- выполняются другие инструкции движения, эта инструкция не может быть выполнена.

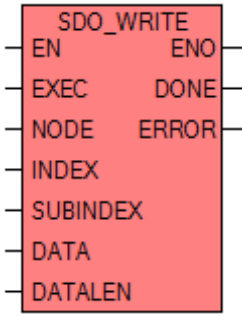
Если EN равен 1, то соответствующая информация об устройстве будет считана в соответствии с параметрами EXECV, EXECS и TIMES.

Если EN равен 0, инструкция не выполняется. Если EN переходит в 0 во время выполнения, инструкция прекратит выполнение.

### 4.5.2 Команды SDO

Команды SDO находятся в группе команд CAN. Их можно использовать совместно с функциями управления движением Kinco или функции главной станции CANOpen. В пользовательском проекте допускается не более 64 инструкций SDO.

#### SDO\_WRITE

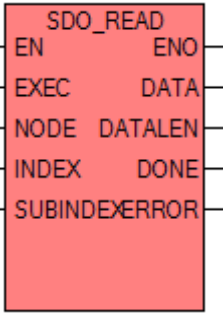
	Название	Применение	Подходит для
LD	SDO_WRITE		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
<i>EXEC</i>	Вход	BOOL	I, Q, V, M, L, SM	Если EN равен 1, команда запускается по переднему фронту EXEC.
<i>NODE</i>	Вход	BYTE	I, Q, V, M, L, SM, константа	ID узла, к которому осуществляется доступ.
<i>INDEX</i>	Вход	WORD	I, Q, V, M, L, SM, константа	Индекс объекта для доступа в OD
<i>SUBINDEX</i>	Вход	BYTE	I, Q, V, M, L, SM, константа	Субиндекс объекта, к которому осуществляется доступ в OD
<i>DATA</i>	Вход	BYTE	I, Q, V, M, L, SM	Начальный адрес байта данных.
<i>DATALEN</i>	Вход	BYTE	I, Q, V, M, L, SM, константа	Длина данных в байтах
<i>DONE</i>	Выход	BOOL	Q, V, M, L, SM	Если SDO выполняется, DONE равен 0; Если SDO закончил связь (ответ получен или тайм-аут), DONE равен 1.
<i>ERROR</i>	Выход	DWORD	Q, V, M, L, SM	Сообщение об ошибке. Смотрите таблицу ниже.

**Примечание:** NODE, INDEX, SUBINDEX, DATALEN должны быть либо все константы, либо все переменные. Параметры DATA и DATALEN составляют блок памяти переменной длины, который должен находиться в допустимых областях памяти, в противном случае результат будет непредсказуемым.

Если EN равен 1, инструкция может быть выполнена по переднему фронту EXEC, выполнение запускается один раз. Если EN равен 0, инструкция не может быть выполнена.

## SDO\_READ

	Название	Применение	Подходит для
LD	SDO_READ		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
<i>EXEC</i>	Вход	BOOL	I, Q, V, M, L, SM	Если EN равен 1, команда запускается по переднему фронту EXEC.
<i>NODE</i>	Вход	BYTE	I, Q, V, M, L, SM, константа	ID узла, к которому осуществляется доступ.

<i>INDEX</i>	Вход	WORD	I, Q, V, M, L, SM, константа	Индекс объекта для доступа в OD
<i>SUBINDEX</i>	Вход	BYTE	I, Q, V, M, L, SM, константа	Субиндекс объекта, к которому осуществляется доступ в OD
<i>DATA</i>	Выход	BYTE	I, Q, V, M, L, SM	Начальный адрес байта данных.
<i>DATALEN</i>	Выход	BYTE	I, Q, V, M, L, SM	Длина данных в байтах
<i>DONE</i>	Выход	BOOL	Q, V, M, L, SM	Если SDO выполняется, DONE равен 0; Если SDO закончил связь (ответ получен или тайм-аут), DONE равен 1.
<i>ERROR</i>	Выход	DWORD	Q, V, M, L, SM	Сообщение об ошибке. Смотрите таблицу ниже.

**Примечание:** NODE, INDEX, SUBINDEX должны быть либо все константы, либо все переменные. Параметры DATA и DATALEN составляют блок памяти переменной длины, который должен находиться в допустимых областях памяти, в противном случае результат будет непредсказуемым.

Если EN равен 1, инструкция может быть выполнена по переднему фронту EXEC, выполнение запускается один раз. Если EN равен 0, инструкция не может быть выполнена.

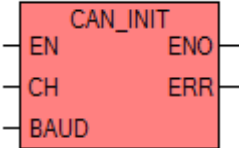
Таблица кодов ошибок для SDO WRITE и SDO READ

Код ошибки	Описание
0	Без ошибки
1	Главная станция не включена
2	Целевой узел не существует
3	Неверные значения входных параметров (например, длина данных)
4	Последняя команда на целевом узле не имеет ответа
5	Буфер отправки или приема ПЛК заполнен
6	Тайм-аут инструкции
7	Получено сообщение об ошибке (не ожидаемое ответное сообщение, ошибка длины и т.д.)
8	Получено сообщение о завершении
9	В этом проекте количество инструкций SDO превысило лимит

### 4.5.3 Команды CAN\_Free

Номер ID сообщений свободной связи не может использовать номер cob-id в протоколе CANOpen!

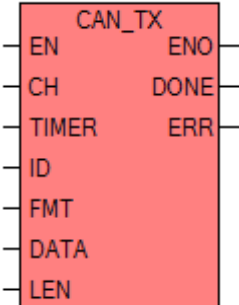
#### CAN\_INIT (инициализация интерфейса CAN)

	Название	Применение	Подходит для
LD	CAN_INIT		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
<i>CH</i>	Вход	INT	константа	CAN интерфейс: 0 для CAN1, 1 для CAN2
<i>BAUD</i>	Вход	INT	V, M, L, константа	Скорость передачи данных 8 --- 1000K 7 --- 800K 6 --- 500K 5 --- 250K 4 --- 125K 3 --- 50K 2 --- 20K 1 --- 10K
<i>ERR</i>	Выход	BOOL	V, M, L, константа	Статус выполнения инструкции: 0 означает успешно, 1 означает ошибку.

Эта инструкция запускается по переднему фронту на входе EN, которая инициализирует указанный интерфейс CAN (CH) и устанавливает скорость передачи данных BAUD.

#### CAN\_TX (автоматическая отправка CAN сообщений)

	Название	Применение	Подходит для
LD	CAN_TX		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
<i>CH</i>	Вход	INT	константа	CAN интерфейс: 0 для CAN1, 1 для CAN2
<i>TIMER</i>	Вход	INT	V, M, L, константа	Период времени для отправки сообщения, единица измерения в мс. 0 означает, что отправка по времени отключена.
<i>ID</i>	Вход	DWORD	V, M, L, константа	ID сообщения для отправки
<i>FMT</i>	Вход	INT	V, M, L, константа	Формат сообщения для отправки. 0 - это стандартный кадр, 1 - расширенный кадр.
<i>DATA</i>	Вход	BYTE	V, M	Первый адрес данных для отправки.
<i>LEN</i>	Вход	BYTE	V, M, L	Длина данных сообщения для отправки в байтах.
<i>DONE</i>	Выход	BOOL	V, M, L	После каждой успешной передачи DONE автоматически устанавливается на 1 и поддерживается как минимум один период сканирования.
<i>ERR</i>	Выход	BOOL	V, M, L	1 означает, что передача не удалась

**Примечание:** параметры ID, FMT, TIMER и LEN должны быть либо все константы, либо все переменные. Параметры DATA и LEN образуют блок памяти переменной длины, который должен находиться в допустимой области памяти, в противном случае результат будет непредсказуем.

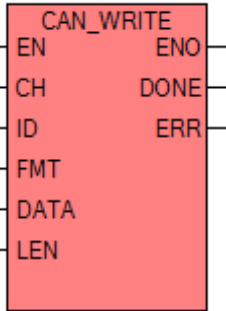
ПЛК поддерживает внутренний список автоматической отправки сообщений. Когда сообщение в таблице удовлетворяет условию отправки, ПЛК автоматически отправит сообщение. Условие автоматической передачи сообщения: если данные в сообщении изменяются, оно сразу же отправляется один раз; в соответствии с установленным временем цикла отправки, отправляется один раз. После отправки сообщения выходной параметр «DONE» автоматически устанавливается на 1 на один цикл сканирования. Если при отправке сообщения происходит сбой (например буфер отправки заполнен или сообщение не может быть отправлено), выходной параметр «ERR» автоматически устанавливается равным 1. Максимальная длина списка отправленных сообщений составляет 48.

Инструкция CAN\_TX используется для добавления сообщения в список сообщений, который определяется идентификатором сообщения, форматом (FMT, указывающий расширенный кадр или стандартный кадр), DATA (DATA, указывающий начальный адрес, где хранится сообщение DATA), и длина LEN. Значение параметра TIMER указывает период времени (мс) для доставки по времени. Если значение TIMER равно 0, это означает, что он не будет отправляться по времени.

Инструкция выполняется по переднему фронту на входе EN. После выполнения этой инструкции ПЛК немедленно добавит сообщение, указанное в инструкции, в список автоматической передачи.

**Следовательно, инструкция CAN\_TX должна выполняться только один раз в проекте.** Кроме того, в проекте может быть задействовано до 48 инструкций CAN\_TX! Общее количество инструкций CAN\_TX и CAN\_WRITE может быть до 64!

## CAN\_WRITE (разовая отправка CAN сообщений)

	Название	Применение	Подходит для
LD	CAN_WRITE		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
<i>CH</i>	Вход	INT	константа	CAN интерфейс: 0 для CAN1, 1 для CAN2
<i>ID</i>	Вход	DWORD	V, M, L, константа	ID сообщения для отправки
<i>FMT</i>	Вход	BYTE	V, M, L, константа	Формат сообщения для отправки. 0 - это стандартный кадр, 1 - расширенный кадр.
<i>DATA</i>	Вход	BYTE	V, M, L	Первый адрес данных для отправки.
<i>LEN</i>	Вход	BYTE	V, M, L, константа	Длина данных сообщения для отправки в байтах.
<i>DONE</i>	Выход	BOOL	V, M, L	После каждой успешной передачи DONE автоматически устанавливается на 1 и поддерживается как минимум один период сканирования.
<i>ERR</i>	Выход	BOOL	V, M, L	1 означает, что передача не удалась

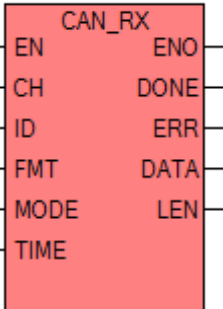
**Примечание:** параметры ID, FMT, TIMER и LEN должны быть либо все константы, либо все переменные. Параметры DATA и LEN образуют блок памяти переменной длины, который должен находиться в допустимой области памяти, в противном случае результат будет непредсказуем.

Сообщение CAN, которое должно быть отправлено, указывается с помощью идентификационного номера, формата (FMT, указывающий кадр расширения или стандартный кадр), DATA (указывающий начальный адрес) и длины LEN.

Эта инструкция выполняется один раз по переднему фронту на входе EN и отправленное сообщение будет записано в буфер отправки внутри ПЛК, а затем отправлено через указанный интерфейс CAN CH посредством планирования ПЛК.

Если инструкция успешно записывает сообщение в буфер отправки, выполнение завершается и DONE будет установлен в 1. Если буфер заполнен (отправка не удалась), инструкция установит DONE и ERR в 1. Общее количество инструкций CAN\_TX и CAN\_WRITE, разрешенных в проекте, составляет до 64!

## CAN\_RX (получение определенного ID сообщения CAN)

	Название	Применение	Подходит для
LD	CAN_RX		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
<i>CH</i>	Вход	INT	константа	CAN интерфейс: 0 для CAN1, 1 для CAN2
<i>ID</i>	Вход	DWORD	V, M, L, константа	ID сообщения для отправки
<i>FMT</i>	Вход	INT	V, M, L, константа	Формат сообщения для отправки. 0 - это стандартный кадр, 1 - расширенный кадр.
<i>MODE</i>	Вход	INT	V, M, L, константа	Режим приема. 0 - режим постоянного приема, 1 - режим разового приема.
<i>TIME</i>	Вход	INT	V, M, L, константа	Время ожидания, единица измерения в мс.
<i>DONE</i>	Выход	BOOL	V, M, L	В режиме разового приема DONE является битом флага успешного приема.
<i>ERR</i>	Выход	BOOL	V, M, L	Бит флага тайм-аута получения.
<i>DATA</i>	Выход	BYTE	V, M	Первый адрес данных, сохраненных в последнем полученном сообщении.
<i>LEN</i>	Выход	BYTE	V, M, L	Длина данных последнего полученного сообщения, единица измерения в байтах.

**Примечание:** параметры ID, FMT, MODE и TIME должны быть либо все константы, либо все переменные. Параметры DATA и LEN образуют блок памяти переменной длины, который должен находиться в допустимой области памяти, в противном случае результат будет непредсказуем.

ПЛК поддерживает список фильтров автоматически принимаемых сообщений. CPU отфильтрует полученные сообщения CAN и примет только сообщения идентификатор и формат которых (стандартный кадр или расширенный кадр), соответствуют значению списка.

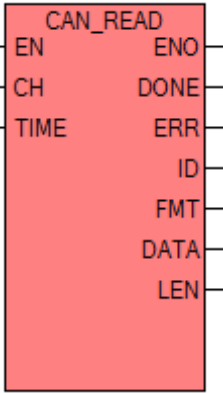
Инструкция CAN\_RX используется для добавления сообщения в список фильтров, который определяется указанным идентификационным номером ID и форматом FMT.

Параметр MODE указывает режим приема. Если MODE равен 1, то это разовый режим приема, инструкция получает указанное сообщение только один раз и завершается после его получения. Если MODE равен 0, это постоянный режим приема и инструкция всегда будет получать указанное сообщение.



Инструкция начинает выполнение по переднему фронту на входе EN. После выполнения этой инструкции указанные значения идентификатора (ID) и формата (FMT) будут немедленно добавлены в список фильтров приема и ПЛК немедленно перейдет в состояние приема, а DONE и ERR будут установлены в 0. Если это разовый режим приема, и если указанное сообщение получено в течение времени TIME, DONE будет установлен в 1 и инструкция выйдет из состояния приема. Если указанное сообщение не получено в течение времени, DONE и ERR будут установлены в 1 и инструкция выйдет из состояния приема. Если это режим постоянного приема, эта инструкция всегда будет отслеживать канал CAN интерфейса и получать все назначенные сообщения после запуска. Если назначенное сообщение не получено за время TIME, после успешного приема, ERR будет установлено на 1. Если назначенное сообщение снова успешно получено, ERR будет установлен в 0. Поэтому в режиме постоянного приема каждая инструкция CAN\_RX должна выполняться только один раз. В проекте может быть использовано до 64 инструкций CAN\_RX!

### CAN\_READ (разовое получение CAN сообщения)

	Название	Применение	Подходит для
LD	CAN_READ		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
CH	Вход	INT	константа	CAN интерфейс: 0 для CAN1, 1 для CAN2
TIME	Вход	INT	V, M, L, константа	После начала приема сообщения, если сообщение не было получено в течение указанного времени, состояние приема завершится, а для ERR будет установлено значение 1.
DONE	Выход	BOOL	V, M, L	DONE установлен на 0 при выполнении, DONE установлен на 1 после отправки.
ERR	Выход	BOOL	V, M, L	Если отправка не удалась (например, буфер отправки переполнен), ERR устанавливается в 1
ID	Выход	DWORD	V, M, L	ID сообщения для получения
FMT	Выход	BYTE	V, M, L	Формат сообщения для отправки. 0 - это стандартный кадр, 1 - расширенный кадр.
DATA	Выход	BYTE	V, M	Начальный адрес байта, куда должны быть отправлены данные.
LEN	Выход	BYTE	V, M, L	Длина данных последнего полученного сообщения, единица измерения в байтах.

**Примечание:** параметры DATA и LEN составляют блок памяти переменной длины, который должен находиться в допустимых областях памяти, в противном случае результат будет непредсказуемым.

Инструкция начинает выполнение по переднему фронту на входе EN: переходит в состояние приема, чтобы получить любое сообщение от указанного CAN-интерфейса CH. Если получите сообщение CAN в течение заданного времени ожидания, ПЛК в соответствии с сообщением установит выходные параметры, такие как ID (идентификационный номер), FMT (формат, обозначающий полученный расширенный кадр или стандартный кадр), DATA (полученный начальный адрес хранилища данных DATA), LEN (длина). Затем, после завершения приема, установит DONE = 1.

Если сообщение не получено в течение указанного периода времени ожидания, полученный статус будет сброшен с DONE и ERR, установленными в 1 оба.

При запуске инструкции CAN\_READ она получит любое сообщение интерфейса CAN, поэтому будьте осторожны, НЕ смешивайте с другими протоколами, такими как CANOpen.

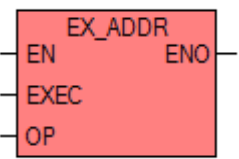
Будьте осторожны при смешивании.

**Примечание:** 1) Инструкция CAN\_READ имеет более низкий приоритет, чем инструкция CAN\_RX.

2) После запуска инструкции CAN\_READ любое сообщение на шине будет получено им.

Следовательно, если в программе вызывается несколько инструкций CAN\_READ, последняя из них вступает в силу.

## EX\_ADDR (изменение конфигурации модуля расширения)

	Название	Применение	Подходит для
LD	EX_ADDR		KS KW103 KW203

Параметр	Вход / Выход	Тип данных	Допустимые области памяти	Описание
EXEC	Вход	BOOL	V, M, L, SM	Бит включения. Если EN равен 1, инструкция включена, что позволяет выполнять.
OP	Вход	INT	V, M, L, константа	Код операции: 181 --- Команда всем модулям расширения, чтобы сохранить их ID и параметры. 99 --- Команда всем модулям расширения, чтобы очистить их ID и параметры.

Эта инструкция отправляет соответствующую команду в модуль расширения на основе значения OP:

- Если значение OP равно 181, модуль расширения автоматически сохранит свой идентификатор и различные параметры (например, форму сигнала, режим фильтрации и т.д.) после получения команды. Когда он включится позже, модуль расширения автоматически прочитает сохраненные данные и перейдет в рабочее состояние, не требующее настройки процессора, поэтому оно может включаться и выключаться независимо от процессора в любое время.
- Если значение OP равно 99, модуль расширения очистит сохраненный идентификатор и параметры после получения команды и будет ждать, пока модуль CPU назначит идентификатор и настроит параметры после повторного включения.

Если EN равен 1, инструкция сканируется и выполнение запускается один раз по переднему фронту EXEC. Если EN равен 0, инструкция не сканируется и не будет выполнена.



<http://systemcontrol.ru/>

2021г.