

Команды IBox контроллера DL06

Приложение к руководству пользователя

Номер руководства DL06-IBOX-S-RUS

Содержание

В этом документе приводятся команды в дополнение к стандартным инструкциям RLL описанным в главе 5 Руководства пользователя DL06. Эти новые команды доступны при использовании приложения DirectSOFT5 для программирования DL06 PLC (для использования новых функций приложения требуется прошивка CPU версии не ниже 2.10). Для дополнительной информации по DirectSOFT5 и получения бесплатной версии приложения заходите на www.automationdirect.com.

Команды работы с аналоговыми данными.....	4
Analog Input/Output Combo Module Pointer Setup (ANLGCMB) (IB-462).....	4
Analog Input Module Pointer Setup (ANLGIN) (IB-460).....	6
Analog Output Module Pointer Setup (ANLGOUT) (IB-461).....	7
Analog Scale 12 Bit BCD to BCD (ANSCL) (IB-423).....	8
Analog Scale 12 Bit Binary to Binary (ANSCLB) (IB-403).....	9
Filter Over Time - BCD (FILTER) (IB-422).....	10
Filter Over Time - Binary (FILTERB) (IB-402).....	11
Hi/Low Alarm - BCD (HILOAL) (IB-421).....	12
Hi/Low Alarm - Binary (HILOALB) (IB-401).....	14
Команды работы с дискретными данными.....	16
Off Delay Timer (OFFDTMR) (IB-302).....	16
On Delay Timer (ONDTMR) (IB-301).....	17
One Shot (ONESHOT) (IB-303).....	18
Push On / Push Off Circuit (PONOFF) (IB-300).....	19
Команды работы с памятью.....	20
Move Single Word (MOVEW) (IB-200).....	20
Move Double Word (MOVED) (IB-201).....	21
Математические команды.....	22
BCD to Real with Implied Decimal Point (BCDTOR) (IB-560).....	22
Double BCD to Real with Implied Decimal Point (BCDTORD) (IB-562).....	23
Math - BCD (MATHBCD) (IB-521).....	24
Math - Binary (MATHBIN) (IB-501).....	25
Math - Real (MATHR) (IB-541).....	26
Real to BCD with Implied Decimal Point and Rounding (RTOBCD) (IB-561).....	27
Real to Double BCD with Implied Decimal Point and Rounding (RTOBCDD) (IB-563).....	28
Square BCD (SQUARE) (IB-523).....	29
Square Binary (SQUAREB) (IB-503).....	30
Square Real (SQUARER) (IB-543).....	31
Sum BCD Numbers (SUMBCD) (IB-522).....	32
Sum Binary Numbers (SUMBIN) (IB-502).....	33
Sum Real Numbers (SUMR) (IB-542).....	34
Коммуникационные команды.....	35
ECOM100 Configuration (ECOM100) (IB-710).....	35
ECOM100 Disable DHCP (ECDHCPD) (IB-736).....	37
ECOM100 Enable DHCP (ECDHCPE) (IB-735).....	39
ECOM100 Query DHCP Setting (ECDHCPQ) (IB-734).....	41
ECOM100 Send E-mail (ECEMAIL) (IB-711).....	43

ECOM100 Restore Default E-mail Setup (ECEMRDS) (IB-713).....	45
ECOM100 E-mail Setup (ECEMSUP) (IB-712)	47
ECOM100 IP Setup (ECIPSUP) (IB-717).....	50
ECOM100 Read Description (ECRDDES) (IB-726).....	52
ECOM100 Read Gateway Address (ECRDGWA) (IB-730).....	54
ECOM100 Read IP Address (ECDIP) (IB-722).....	56
ECOM100 Read Module ID (ECRDMID) (IB-720)	58
ECOM100 Read Module Name (ECRDNAM) (IB-724)	60
ECOM100 Read Subnet Mask (ECRDSNM) (IB-732)	62
ECOM100 Write Description (ECWRDES) (IB-727).....	64
ECOM100 Write Gateway Address (ECWRGWA) (IB-731)	66
ECOM100 Write IP Address (ECWRIP) (IB-723).....	68
ECOM100 Write Module ID (ECWRMID) (IB-721).....	70
ECOM100 Write Name (ECWRNAM) (IB-725)	72
ECOM100 Write Subnet Mask (ECWRSNM) (IB-733)	74
ECOM100 RX Network Read (ECRX) (IB-740)	76
ECOM100 WX Network Write(ECWX) (IB-741)	78
NETCFG Network Configuration (NETCFG) (IB-700).....	80
Network RX Read (NETRX) (IB-701)	81
Network WX Write (NETWX) (IB-702)	83
Команды работы с модулем H0-CTRIO.....	85
CTRIO Configuration (CTRIO) (IB-1000).....	85
CTRIO Add Entry to End of Preset Table (CTRADPT) (IB-1005).....	87
CTRIO Clear Preset Table (CTRCLRT) (IB-1007).....	89
CTRIO Edit Preset Table Entry (CTREDPT) (IB-1003)	91
CTRIO Edit Preset Table Entry and Reload (CTREDRL) (IB-1002).....	94
CTRIO Initialize Preset Table (CTRINPT) (IB-1004)	96
CTRIO Initialize Preset Table on Reset (CTRINTR) (IB-1010)	98
CTRIO Load Profile (CTRLDPR) (IB-1001)	100
CTRIO Read Error (CTRORDER) (IB-1014)	102
CTRIO Run to Limit Mode (CTRRTLM) (IB-1011)	104
CTRIO Run to Position Mode (CTRRTPM) (IB-1012)	106
CTRIO Velocity Mode (CTRVELO) (IB-1013).....	108
CTRIO Write File to ROM (CTRWFTR) (IB-1006).....	110

Команды работы с аналоговыми данными

Analog Input/Output Combo Module Pointer Setup (ANLGCMB) (IB-462)

На первом скан-цикле контроллера, Analog Input/Output Combo Module Pointer Setup IBox настроит CPU на работу с комбинированным модулем аналогового ввода/вывода.

Команда ANLGCMB IBox задает формат данных и адрес модуля в зависимости от типа CPU: номер каркаса (Base#) и номер модуля в каркасе (Slot#).

Input Data Address – это адрес первой ячейки V-памяти DL06, начиная с которой будут сохраняться полученные со входов аналоговые значения. Одна ячейка на каждый входной канал данных.

Output Data Address – это адрес первой ячейки V-памяти, начиная с которой будут помещаться аналоговые значения полученные программной логикой или внешними устройствами. Одна ячейка на каждый выходной канал данных.

Этот IBox срабатывает автоматически на первом скане, поэтому не может иметь на входе дополнительной релейной логики.

Так как IBox настраивает только один модуль, вам необходимо использовать столько IBox, сколько таких модулей есть в контроллере.

Параметры:

- **Base # (K0-Local):** номер каркаса (Base), должен быть 0 для DL06.
- **Slot #:** номер слота в каркасе, где расположен аналоговый модуль.
- **Number of Input Channels:** сколько каналов должен опрашивать модуль.
- **Input Data Format (0-BCD 1-BIN):** формат данных, используемый для хранения аналоговых значений. Все каналы будут использовать этот формат.
- **Input Data Address:** начальный адрес блока V-памяти, используемого для хранения аналоговых данных
- **Number of Output Channels:** со сколькими каналами аналогового вывода должен работать
- **Output Data Format (0-BCD 1-BIN):** Формат данных, используемый для хранения значений аналоговых выходов. Все каналы будут использовать этот формат.
- **Output Data Address:** начальный адрес блока V-памяти, используемого для хранения аналоговых выходов.

Параметр	Диапазон DL06
Base # (K0-Local) K	Константа: K0 (только локальный каркас)
Slot # K	Константа: K1
Number of Input Channels K	Константа: K1-8
Input Data Format (0-BCD 1-BIN) K	Константа: K0-1
Input Data Address V	См. карту памяти DL06 - Слова данных
Number of Output Channels K	Константа: K1-8
Output Data Format (0-BCD 1-BIN) K	Константа: K0-1
Output Data Address V	См. карту памяти DL06 - Слова данных

Пример:

DirectSOFT5 Команды контроллера DL06

В этом примере команда ANLGCMB используется для настройки комбинированного модуля аналоговых входов-выходов, установленного в слот 2. Опрашиваются 4 аналоговых входа, получаемые данные записываются в ячейки V2000-V2003 в BCD формате. Задействованы два выходных канала, аналоговые значения берутся из ячеек V2100-V2101 в BCD формате. С этой командой не используется входная логика.

1	Analog Input/Output Combo Module Pointer Setup	
	ANLGCMB	IB-462
	Base # (K0-Local)	K0
	Slot #	K2
	Number of Input Channels	K4
	Input Data Format (0-BCD 1-BIN)	K0
	Input Data Address	V2000
	Number of Output Channels	K2
	Output Data Format (0-BCD 1-BIN)	K0
	Output Data Address	V2100

Analog Input Module Pointer Setup (ANLGIN) (IB-460)

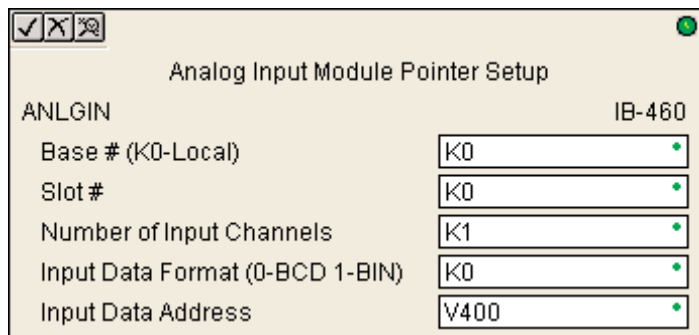
На первом скан-цикле контроллера этот IBox настроит CPU на работу с модулем аналогового ввода.

Команда ANLGIN задает формат данных и адрес модуля в зависимости от типа CPU: номер каркаса (Base#) и номер модуля в каркасе (Slot#).

Input Data Address – это адрес первой ячейки V-памяти, начиная с которой будут сохраняться полученные со входов аналоговые значения. Одна ячейка на каждый входной канал данных.

Этот IBox срабатывает автоматически на первом скане, поэтому не может иметь на входе дополнительной релейной логики.

Так как IBox настраивает только один модуль, вам необходимо использовать столько IBox, сколько таких модулей есть в контроллере.



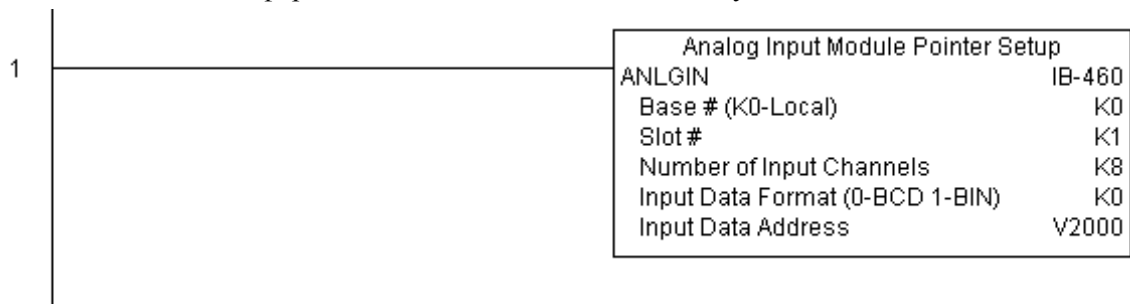
Параметры:

- **Base # (K0-Local):** номер каркаса (Base), должен быть 0 для DL06.
- **Slot #:** номер слота в каркасе, где расположен аналоговый модуль.
- **Number of Input Channels:** сколько каналов должен опрашивать модуль.
- **Input Data Format (0-BCD 1-BIN):** формат данных, используемый для хранения аналоговых значений. Все каналы будут использовать этот формат.
- **Input Data Address:** начальный адрес блока V-memory, используемого для хранения аналоговых данных

Параметр	Диапазон DL06
Base # (K0-Local) K	Константа: K0 (только локальный каркас)
Slot # K	Константа: K1-4
Number of Input Channels K	Константа: K1-8
Input Data Format (0-BCD 1-BIN) K	Константа: K0-1
Input Data Address V	См. карту памяти DL06 - Слова данных

Пример:

В этом примере команда ANLGIN используется для настройки модуля аналоговых входов, установленного в слот 1. Опрашиваются 8 аналоговых входов, получаемые данные записываются в ячейки V2000-V2007 в BCD формате. С этой командой не используется входная логика.



Analog Output Module Pointer Setup (ANLGOUT) (IB-461)

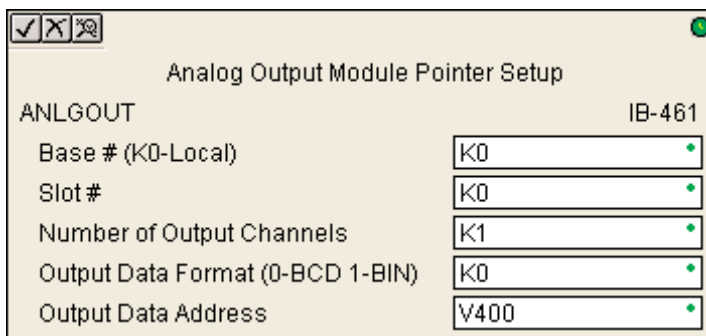
На первом скан-цикле контроллера этот IBox настроит CPU на работу с модулем аналогового вывода.

Команда ANLGOUT IBox задает формат данных и адрес модуля в зависимости от типа CPU: номер каркаса (Base#) и номер модуля в каркасе (Slot#).

Output Data Address – это адрес первой ячейки V-памяти, начиная с которой будут помещаться аналоговые значения полученные программной логикой или внешними устройствами. Одна ячейка на каждый выходной канал данных.

Этот IBox срабатывает автоматически на первом скане, поэтому не может иметь на входе дополнительной релейной логики.

Так как IBox настраивает только один модуль, вам необходимо использовать столько IBox, сколько таких модулей есть в контроллере.



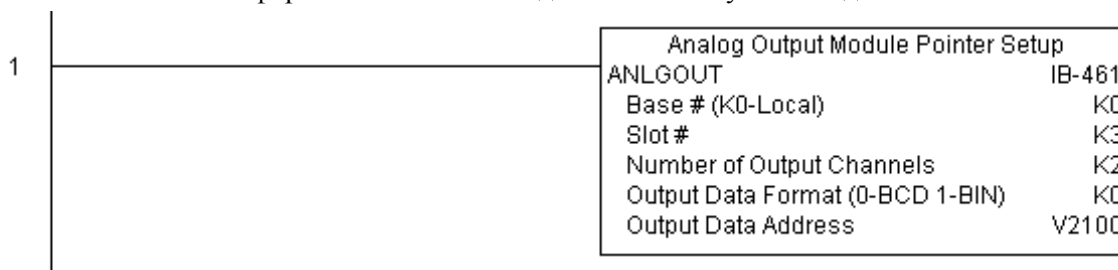
Параметры:

- **Base # (K0-Local):** номер каркаса (Base), должен быть 0 для DL06.
- **Slot #:** номер слота в каркасе, где расположен аналоговый модуль.
- **Number of Output Channels:** со сколькими каналами аналогового вывода должен работать
- **Output Data Format (0-BCD 1-BIN):** формат данных, используемый для хранения значений аналоговых выходов. Все каналы будут использовать этот формат.
- **Output Data Address:** начальный адрес блока V-памяти, используемого для хранения аналоговых выходов.

Параметр	Диапазон DL06
Base # (K0-Local) K	Константа: K0 (только локальный каркас)
Slot # K	Константа: K0-7
Number of Output Channels K	Константа: K1-8
Output Data Format (0-BCD 1-BIN) K	Константа: K0-1
Output Data Address V	См. карту памяти DL06 - Слова данных

Пример:

В этом примере команда ANLGOUT используется для настройки модуля аналоговых выходов, установленного в слот 3. Задействованы два выходных канала, аналоговые значения берутся из ячеек V2100-V2101 в BCD формате. С этой командой не используется входная логика.



Analog Scale 12 Bit BCD to BCD (ANSCL) (IB-423)

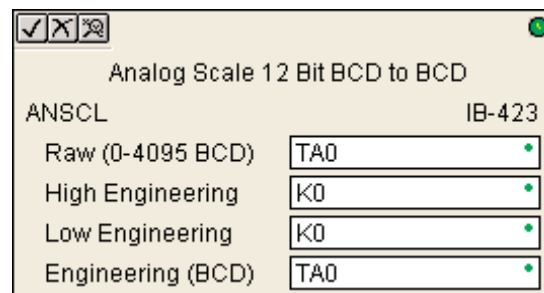
Этот IBox масштабирует 12-битовое BCD значение (0-4095) в BCD значение в технических (инженерных) единицах. Указываются: максимальное инженерное значение (при исходном Raw = 4095), минимальное инженерное значение (при исходном Raw = 0), адрес Ячейки V-памяти для результата.

Формула преобразования:

$$Engineering = (Raw * ((High Eng - Low Eng) / 4095))$$

Результат в формате BCD и может быть в диапазоне от 0 до 9999. Для масштабирования двоичных значений см. команду ANSCLB.

Этот IBox работает только с беззнаковыми исходными величинами. Он не работает с биполярными величинами и числами в формате «знак плюс значение».



Параметры:

- **Raw (0-4095 BCD):** адрес ячейки памяти с исходным 12-битовым BCD значением для масштабирования. Беззнаковое однополярное число от 0 до 4095.
- **High Engineering:** верхний предел шкалы в инженерных единицах, соответствующий исходному (Raw) = 4095.
- **Low Engineering:** предел шкалы в инженерных единицах, соответствующий исходному (Raw) = 0.
- **Engineering (BCD):** адрес ячейки памяти для значения в инженерных единицах.

Параметр	Диапазон DL06
Raw (0-4095 BCD) V,P	См. карту памяти DL06 - Слова данных
High Engineering K	Константа: K0-9999
Low Engineering K	Константа: K0-9999
Engineering (BCD) V,P	См. карту памяти DL06 - Слова данных

Пример:

В этом примере инструкция ANSCL используется для масштабирования исходного (например измеренного) значения в ячейке V2000 в BCD формате. Инженерная шкала от 0 до 100. Расчетное значение в инженерных единицах помещается в ячейку V2100 в BCD формате.

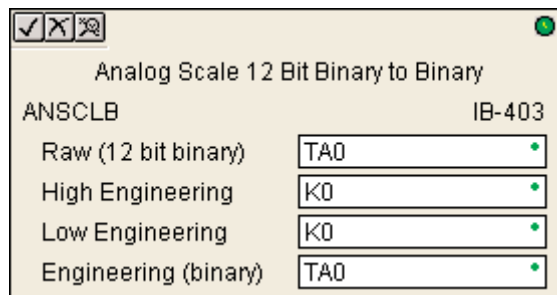


Analog Scale 12 Bit Binary to Binary (ANSCLB) (IB-403)

Этот IBox масштабирует 12-битовое двоичное значение (0-4095) в двоичное значение в технических (инженерных) единицах. Указываются: максимальное инженерное значение (при исходном Raw = 4095), минимальное инженерное значение (при исходном Raw = 0), адрес Ячейки V-памяти для результата.

Результат в двоичном формате и может быть в диапазоне от 0 до 65535. Для масштабирования значений в формате BCD см. команду ANSCL.

Этот IBox работает только с беззнаковыми исходными величинами. Он не работает с биполярными величинами и числами в формате «знак плюс значение».



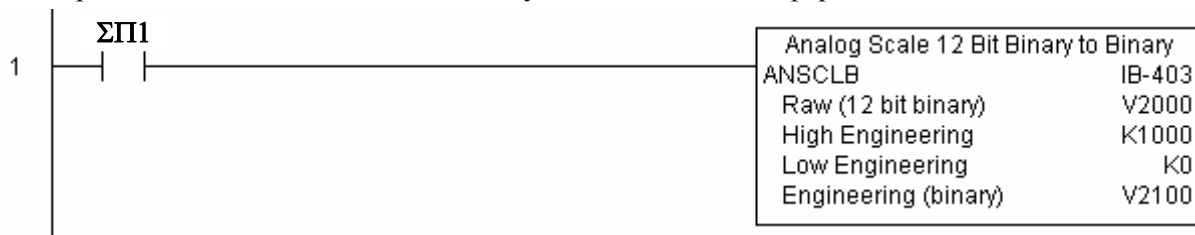
Параметры:

- **Raw (12 bit binary):** адрес ячейки памяти с исходным 12-битовым двоичным значением для масштабирования. Беззнаковое однополярное двоичное десятичное число от 0 до 4095.
- **High Engineering:** верхний предел шкалы в инженерных единицах, соответствующий исходному (Raw) = 4095.
- **Low Engineering:** предел шкалы в инженерных единицах, соответствующий исходному (Raw) = 0.
- **Engineering (binary):** адрес ячейки памяти для двоичного десятичного значения в инженерных единицах.

Параметр	Диапазон DL06
Raw (12 bit binary) V,P	См. карту памяти DL06 - Слова данных
High Engineering K	Константа: K0-65535
Low Engineering K	Константа: K0-65535
Engineering (binary) V,P	См. карту памяти DL06 - Слова данных

Пример:

В этом примере инструкция ANSCLB используется для масштабирования исходного (например измеренного) двоичного значения в ячейке V2000. Инженерная шкала от 0 до 1000. Расчетное значение в инженерных единицах помещается в ячейку V2100 в двоичном формате.



Filter Over Time - BCD (FILTER) (IB-422)

Реализует фильтр первого порядка для указанного исходного BCD значения на задаваемом временном интервале.

Формула расчета: $New = Old + ((Raw - Old) / FDC)$,

где:

New – новое фильтрованное значение,

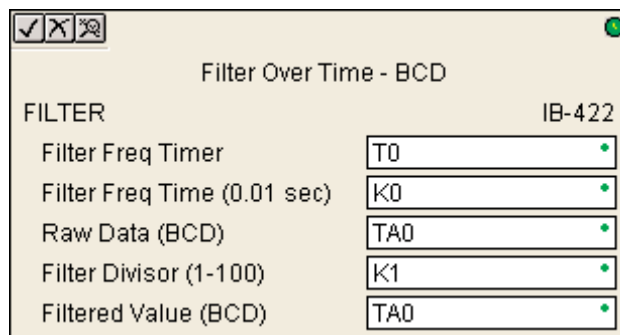
Old – старое фильтрованное значение,

FDC – постоянная фильтрации,

Raw – исходное число.

Степень сглаживания задается константой FDC (Filter Divisor Constant), это целое число в диапазоне 1-100. При FDC=1 фильтрации нет, при FDC=100 фильтрация максимальная.

Частота выполнения вычислений (временной интервал сглаживания) указывается в сотых секунды (0.01 сек.) в параметре Filter Freq Time.



Обратите внимание.

В этот IBox встроена инструкция работы с заданным таймером, эта инструкция НЕ ДОЛЖНА БЫТЬ использована ГДЕ-ЛИБО В ПРОГРАММЕ еще.

Логическое условие исполнения этой команды определяет будет ли выполняться вычисление. Если условие ложно, результат фильтрации Filter Value обновляться не будет.

При первом скане (при запуске программы) Filter Value инициализируется нулем.

Параметры:

- **Filter Frequency Timer**: номер таймера, используемого для фильтрации.
- **Filter Frequency Time (0.01sec)**: частота вычислений
- **Raw Data (BCD)**: адрес ячейки исходного значения в BCD формате.
- **Filter Divisor (1-100)**: целая константа задает степень фильтрации. Большее значение усиливает сглаживание. При значении равном 1 фильтрации нет.
- **Filtered Value (BCD)**: адрес ячейки фильтрованного значения в двоичном формате.

Параметр	Диапазон DL06
Filter Frequency Timer T	Номер таймера: T0-377
Filter Frequency Time (0.01 sec) K	Константа: K0-9999
Raw Data (BCD) V	См. карту памяти DL06 - Слова данных
Filter Divisor (1-100) K	Константа: K1-100
Filtered Value (BCD) V	См. карту памяти DL06 - Слова данных

Пример:

Здесь команда FILTER используется для фильтрации ячейки V2000 в BCD формате. Задана частота вычислений таймером T0 равной раз в 0.5 сек. Константа фильтрации равна 2. Фильтрованное значение помещается в ячейку V2100.



Filter Over Time - Binary (FILTERB) (IB-402)

Реализует фильтр первого порядка для указанного исходного двоичного значения на задаваемом временном интервале. Степень сглаживания задается константой FDC (Filter Divisor Constant). При FDC=1 фильтрации нет, FDC=100 фильтрация максимальная.

Формула расчета: $New = Old + ((Raw - Old) / FDC)$.

New – новое фильтрованное значение,

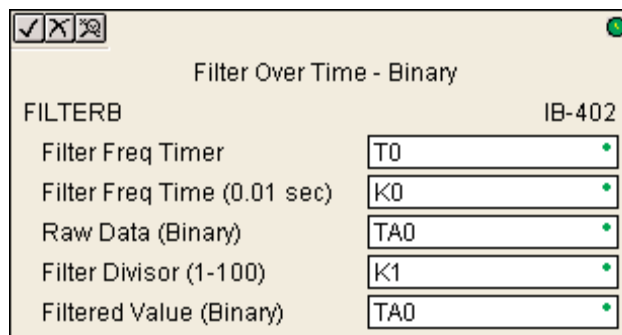
Old – старое фильтрованное значение,

FDC – постоянная фильтрации,

Raw – исходное число.

Степень сглаживания задается константой FDC (Filter Divisor Constant), это целое число в диапазоне 1-100. При FDC=1 фильтрации нет, при FDC=100 фильтрация максимальная.

Частота выполнения вычислений (временной интервал сглаживания) указывается в сотых секунды (0.01 сек.) в параметре Filter Freq Time.



Обратите внимание.

В этот IBox встроена инструкция работы с заданным таймером, эта инструкция НЕ ДОЛЖНА БЫТЬ использована ГДЕ-ЛИБО В ПРОГРАММЕ еще.

Логическое условие исполнения этой команды определяет будет ли выполняться вычисление. Если условие ложно, результат фильтрации Filter Value обновляться не будет.

При первом скане (при запуске программы) Filter Value инициализируется нулем.

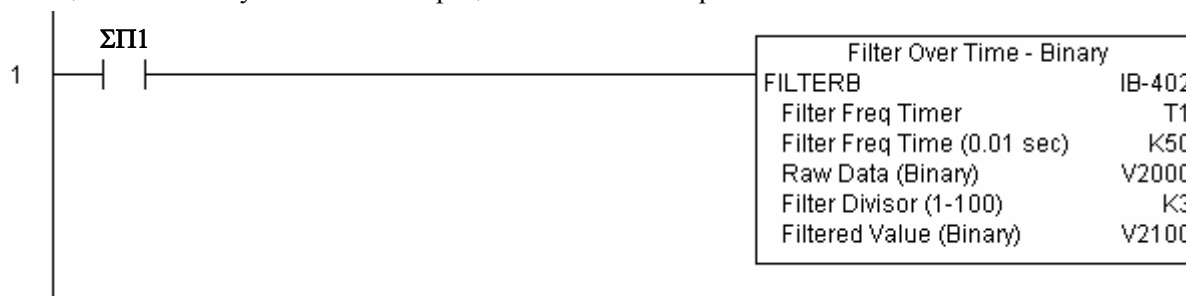
Параметры:

- **Filter Frequency Timer:** номер таймера, используемого для фильтрации.
- **Filter Frequency Time (0.01sec):** частота вычислений
- **Raw Data (Binary):** адрес ячейки исходного значения в двоичном формате.
- **Filter Divisor (1-100):** целая константа задает степень фильтрации. Большее значение усиливает сглаживание. При значении равном 1 фильтрации нет.
- **Filtered Value (Binary):** адрес ячейки фильтрованного значения в двоичном формате.

Параметр	Диапазон DL06
Filter Frequency Timer T	Номер таймера: T0-T377
Filter Frequency Time (0.01 sec) K	Константа: K0-9999
Raw Data (Binary) V	См. карту памяти DL06 - Слова данных
Filter Divisor (1-100) K	Константа: K1-100
Filtered Value (Binary) V	См. карту памяти DL06 - Слова данных

Пример:

Здесь команда FILTERB используется для фильтрации ячейки V2000 в двоичном формате. Задана частота вычислений таймером T1 равной раз в 0.5 сек. Константа фильтрации равна 3. Фильтрованное значение помещается в ячейку V2100. Фильтрация выполняется при бите C100 = 1.



Hi/Low Alarm - BCD (HILOAL) (IB-421)

Эта команда сравнивает BCD значение в V-memory с предельными значениями и взводит соответствующие биты состояния. При вводе предельных значений, надо чтобы High-High limit \geq High limit $>$ Low limit \geq Low-Low limit.

Когда срабатывают High-High или Low-Low, то устанавливаются также и соответствующие им High или Low.

Биты сигнализирующие отклонения должны быть уникальными. Нельзя использовать один и тот же бит для сигнализации нескольких отклонений.

Если логическое условие исполнения этой команды станет ложным, то все биты состояния будут сброшены вне зависимости от значения контролируемого параметра.

HILOAL	IB-421
Monitoring Value (BCD)	TA0 •
High-High Limit	TA0 •
High-High Alarm	C0 •
High Limit	TA0 •
High Alarm	C0 •
Low Limit	TA0 •
Low Alarm	C0 •
Low-Low Limit	TA0 •
Low-Low Alarm	C0 •

Параметры:

- **Monitoring Value (BCD):** адрес ячейки, содержимое (в формате BCD) которой контролируется.
- **High-High Limit:** предельное значение High-High в BCD формате
- **High-High Alarm:** бит достижения значения High-High
- **High Limit:** предельное значение High в BCD формате
- **High Alarm:** бит достижения значения High
- **Low Limit:** предельное значение Low в BCD формате
- **Low Alarm:** бит достижения значения Low
- **Low-Low Limit:** предельное значение Low- Low в BCD формате
- **Low-Low Alarm:** бит достижения значения Low-Low

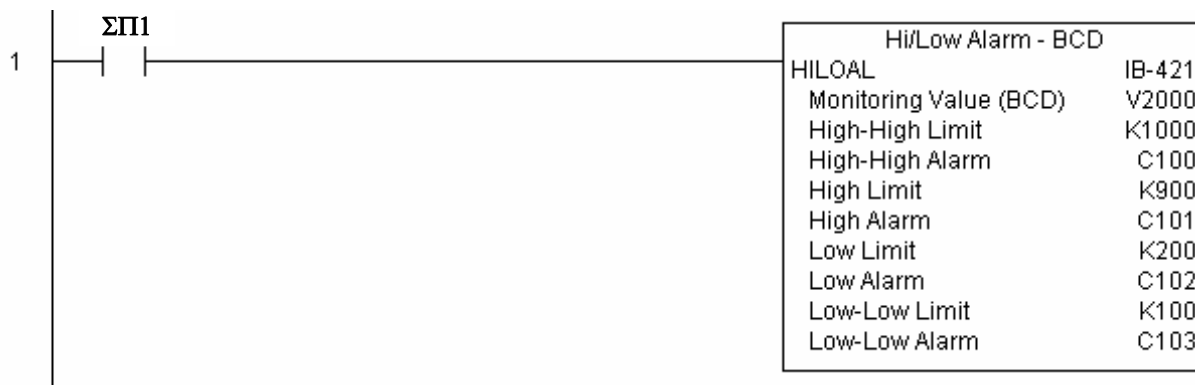
Параметр	Диапазон DL06
Monitoring Value (BCD) V	См. карту памяти DL06 - Слова данных
High-High Limit V, K	Константа K0-K9999 или См. карту памяти DL06 - Слова данных
High-High Alarm X, Y, C, GX,GY, B	См. карту памяти DL06 - Слова данных
High Limit V, K	Константа K0-K9999 или См. карту памяти DL06 - Слова данных
High Alarm. X, Y, C, GX,GY, B	См. карту памяти DL06 - Слова данных
Low Limit V, K	Константа K0-K9999 или См. карту памяти DL06 - Слова данных
Low Alarm X, Y, C, GX,GY,B	См. карту памяти DL06 - Слова данных
Low-Low Limit V, K	Константа K0-K9999 или См. карту памяти DL06 - Слова данных
Low-Low Alarm X, Y, C, GX,GY, B	См. карту памяти DL06 - Слова данных

Пример:

Здесь команда HILOAL используется для контроля значения ячейки V2000 заданного в BCD формате. Если значение достигает верхнего предела 900, то устанавливается бит C101. Если значение продолжает повышаться и достигает предела High-High, то также устанавливается бит C100. В этом случае будут установлены оба бита. Оба предела High-High и High могут быть установлены на одно значение.

DirectSOFT5 Команды контроллера DL06

Если значение ячейки V2000 достигает или опускается ниже предела 200, то устанавливается бит C102. А при снижении ниже 100 установится еще и C103. В этом случае будут установлены оба бита. Пределы Low-Low и Low могут быть установлены на одно значение.



Hi/Low Alarm - Binary (HILOALB) (IB-401)

Эта команда сравнивает двоичное значение в ячейке V-методу с предельными значениями и устанавливает соответствующие биты состояния. При вводе предельных значений, надо чтобы High-High limit \geq High limit $>$ Low limit \geq Low-Low limit.

Когда срабатывают High-High или Low-Low, то устанавливаются также и соответствующие им High или Low биты.

Биты сигнализирующие отклонения должны быть уникальными. Нельзя использовать один и тот же бит для сигнализации нескольких отклонений.

Если логическое условие выполнения этой команды станет ложным, то все биты состояния будут сброшены вне зависимости от значения контролируемого параметра.

Параметры:

- **Monitoring Value (Binary):** адрес ячейки, содержимое (в двоичном формате) которой контролируется.
- **High-High Limit:** предельное значение High-High в двоичном формате
- **High-High Alarm:** бит достижения значения High-High
- **High Limit:** предельное значение High в двоичном формате
- **High Alarm:** бит достижения значения High
- **Low Limit:** предельное значение Low в двоичном формате
- **Low Alarm:** бит достижения значения Low
- **Low-Low Limit:** предельное значение Low- Low в двоичном формате
- **Low-Low Alarm:** бит достижения значения Low-Low

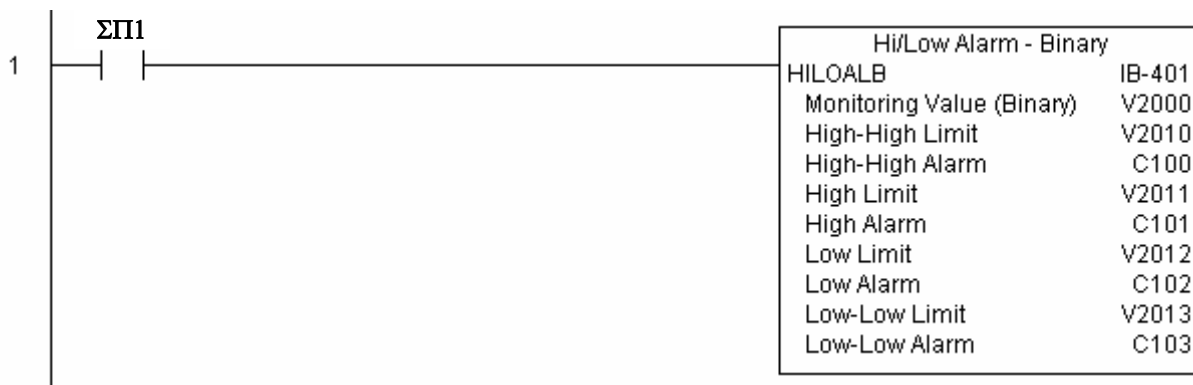
Параметр	Диапазон DL06
Monitoring Value (Binary) V	См. карту памяти DL06 - Слова данных
High-High Limit V, K	Конст. K0-K65535 или См. карту памяти DL06 - Слова данных
High-High Alarm X, Y, C, GX,GY, B	См. карту памяти DL06 - Слова данных
High Limit V, K	Конст. K0-K65535 или См. карту памяти DL06 - Слова данных
High Alarm. X, Y, C, GX,GY, B	См. карту памяти DL06 - Слова данных
Low Limit V, K	Конст. K0-K65535 или См. карту памяти DL06 - Слова данных
Low Alarm X, Y, C, GX,GY,B	См. карту памяти DL06 - Слова данных
Low-Low Limit V, K	Конст. K0-K65535 или См. карту памяти DL06 - Слова данных
Low-Low Alarm X, Y, C, GX,GY, B	См. карту памяти DL06 - Слова данных

Пример:

Здесь команда HILOALB используется для контроля значения ячейки V2000 заданного в двоичном формате. Если значение достигает верхнего предела, заданного двоичной ячейкой V2011, то устанавливается бит C101. Если значение продолжает повышаться и достигает предела High-High, заданного двоичной ячейкой V2010, то также устанавливается бит C100. В этом случае будут установлены оба бита. Оба предела High-High и High могут быть установлены на одно значение.

DirectSOFT5 Команды контроллера DL06

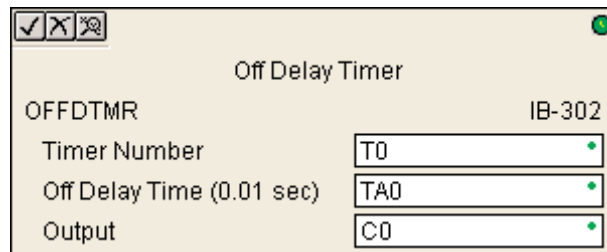
Если значение ячейки V2000 достигает или опускается ниже предела заданного ячейкой V2012, то устанавливается бит C102. А при снижении ниже значения заданного ячейкой V2012 установится еще и C103. В этом случае будут установлены оба бита. Пределы Low-Low и Low могут быть установлены на одно значение.



Команды работы с дискретными данными

Off Delay Timer (OFFDTMR) (IB-302)

Задержка сброса бита. Этот IBox осуществляет задержку отключения (OFF) указанного выхода на заданное в таймере значение. Когда вход переходит в состояние ON, выход тоже переходит в ON. Когда на входе появляется OFF, выход остается в ON на время работы таймера и после его истечения выход переходит в OFF. Если вход перейдет в состояние ON до истечения времени, таймер будет сброшен и выход останется в состоянии ON. В следующий раз (вход=OFF) таймер запустится сначала.



Эта команда для задержки использует таймер. Таймер с этим же номером не должен быть использован в другом месте программы пользователя.

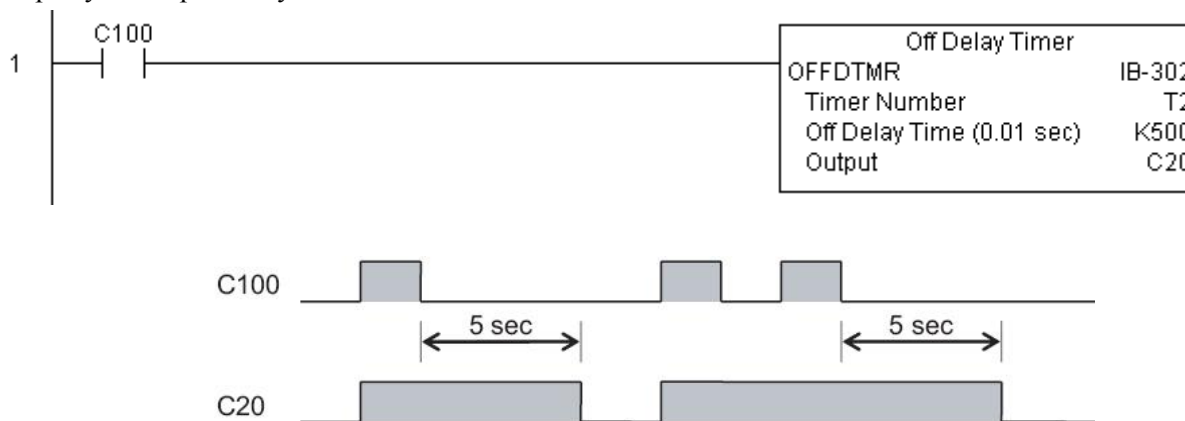
Параметры:

- **Timer Number:** номер используемого таймера.
- **Off Delay Time (0.01sec):** задержка времени в сотых секунды на сброс выхода
- **Output:** выход, которым управляет таймер.

Параметр	Диапазон DL06
Timer Number T	Таймер: T0-T377
Off Delay Time K,V	Конст. K0-K9999 или См. карту памяти DL06 - Слова данных
Output X, Y, C, GX,GY, B	См. карту памяти DL06

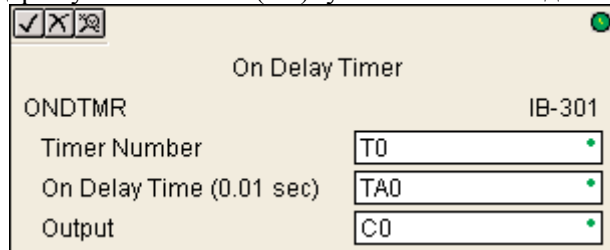
Пример:

В этом примере используется инструкция OFFDTMR для задержки сброса бита C20. Выбранный таймер T2 настроен на выдержку 5 секунд. Когда C100 устанавливается (On), C20 также устанавливается и будет оставаться в этом состоянии пока C100=On. Если C100 сбросится, то C20 также сбросится, но через задержку таймера 5 секунд.



On Delay Timer (ONDTMR) (IB-301)

Задержка установки бита. Этот IBox осуществляет задержку включения (On) указанного выхода на заданное в таймере значение. Когда вход переходит в состояние Off, выход тоже переходит в Off. Когда на входе появляется ON, выход остается в OFF на время работы таймера и после его истечения выход переходит в ON. Если вход перейдет в состояние OFF до истечения времени, таймер будет сброшен и выход останется в состоянии OFF. В следующий раз (вход=ON) таймер запустится сначала.



Эта команда для задержки использует таймер. Таймер с этим же номером не должен быть использован в другом месте программы пользователя.

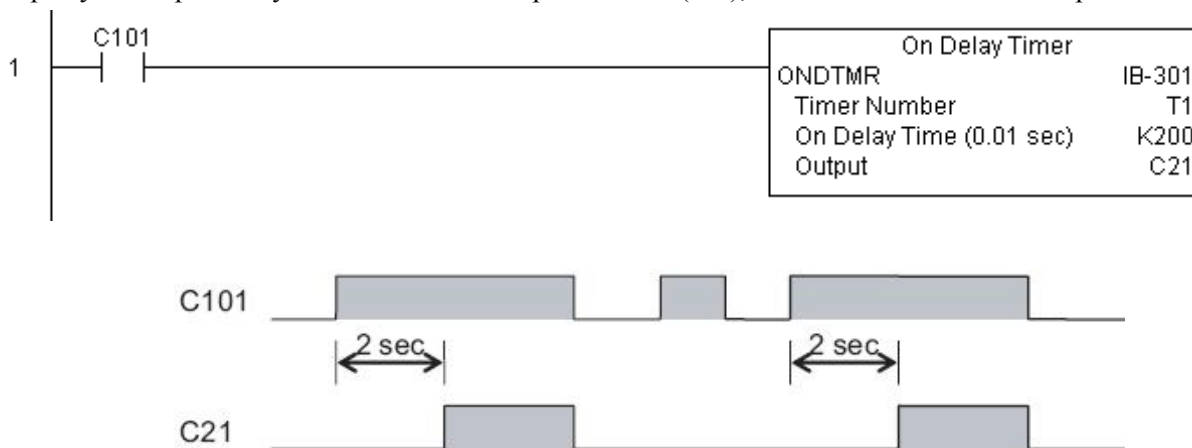
Параметры:

- **Timer Number:** номер используемого таймера.
- **ON Delay Time (0.01sec):** задержка времени в сотых секунды на сброс выхода
- **Output:** выход, которым управляет таймер.

Параметр	Диапазон DL06
Timer Number T	Таймер: T0-T377
ON Delay Time K,V	Конст. K0-K9999 или См. карту памяти DL06 - Слова данных
Output X, Y, C, GX,GY, B	См. карту памяти DL06

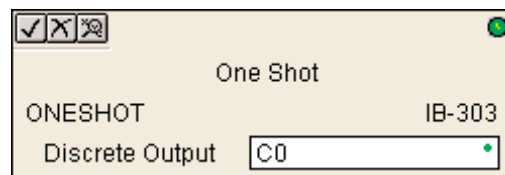
Пример:

В этом примере используется инструкция ONDTMR для задержки установки бита C21. Выбранный таймер T1 настроен на выдержку 2 секунд. Когда C101 устанавливается (On), C21 устанавливается через задержку таймера 2 секунды. Когда C101 сбрасывается (Off), C21 также немедленно сбрасывается.



One Shot (ONESHOT) (IB-303)

Этот IBox будет переводить указанный бит в состояние ON на один скан при переходе входа IBox из OFF в ON. Этот IBox выполняет тоже, что и PD Coil (Positive Differential) только по-другому назван.



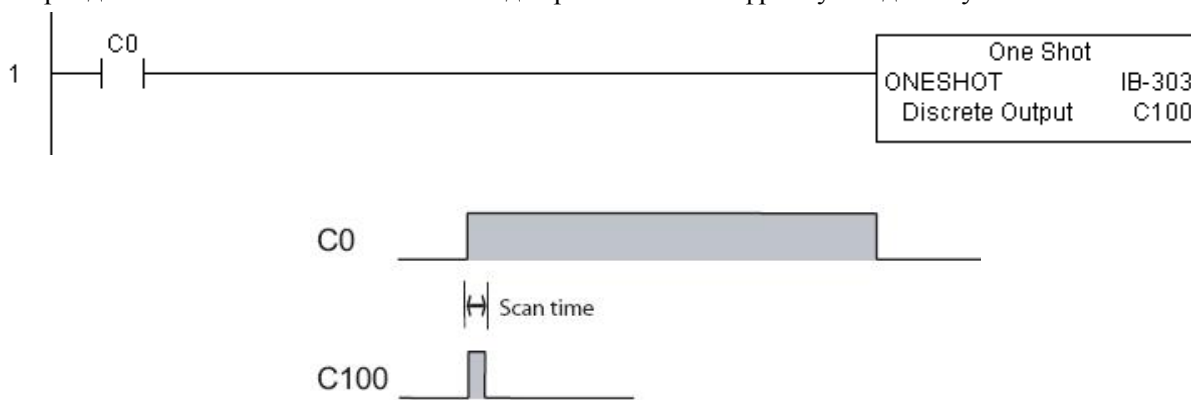
Параметры:

- **Discrete Output:** выход, активируемый на время одного скан-цикла.

Параметр	Диапазон DL06
Discrete Output X, Y, C	См. карту памяти DL06

Пример:

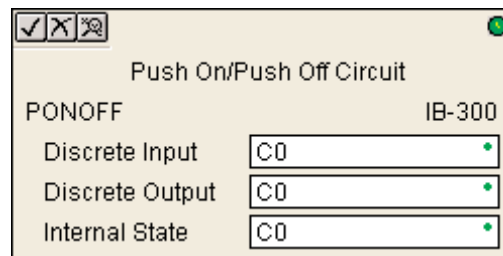
Здесь команда ONESHOT используется для возведения бита C100 на один скан-цикл после того как бит C0 перейдет из состояния Off в On. Команда срабатывает по фронту входного условия.



Push On / Push Off Circuit (PONOFF) (IB-300)

Эта команда будет переключать указанный дискретный выход каждый раз, когда указанный вход будет переходить из состояния OFF в состояние ON.

Требуется дополнительный бит, сохраняющий внутреннее состояние. Этот бит НЕ ДОЛЖЕН БЫТЬ использован где-либо еще в программе.



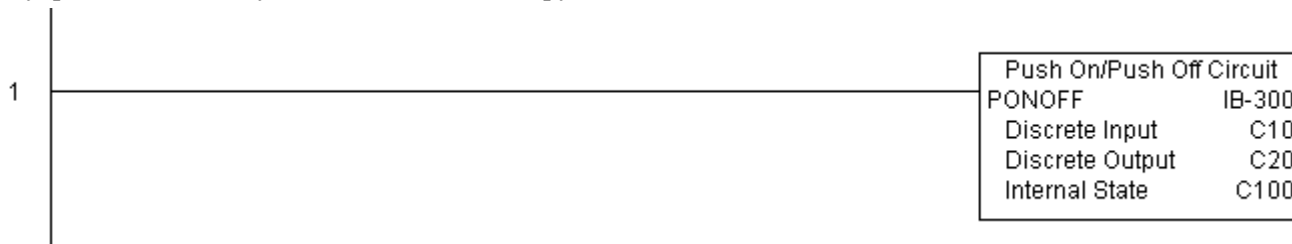
Парметры:

- **Discrete Input:** адрес входа.
- **Discrete Output:** адрес выхода, который будет переключаться при истинном значении входа.
- **Internal State:** бит хранения внутреннего состояния.

Параметр	Диапазон DL06
Discrete Input X,Y,C,S,T,CT,GX,GY,SP,B,PB	См. карту памяти DL06
Discrete Output X,Y,C,GX,GY,B	См. карту памяти DL06
Internal State X, Y, C	См. карту памяти DL06

Пример:

Здесь команда PONOFF применяется для управления состоянием бита C20 в зависимости от бита C10. Когда C10 становится On в первый раз, C20 устанавливается (On). Затем C10 сбрасывается, при этом C20 не изменяется. Когда C20 становится On в следующий раз, C20 сбрасывается (Off). Бит C100 – внутренний, используется только этой инструкцией.



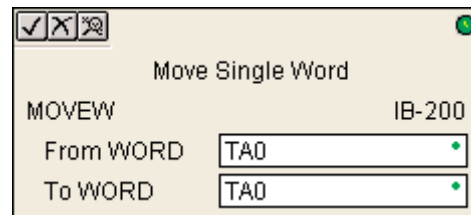
Команды работы с памятью

Move Single Word (MOVEW) (IB-200)

Эта команда копирует слово данных из одной ячейки памяти в другую либо записывает шестнадцатеричную константу в ячейку памяти.

Параметры:

- **From WORD:** адрес ячейки памяти или константа – источник данных.
- **To WORD:** адрес ячейки памяти – приемник данных.



Параметр	Диапазон DL06
From WORD V,P,K	Конст. K0-FFFF, См. карту памяти DL06: Слова данных
To WORD V,P	См. карту памяти DL06: Слова данных

Пример:

В этом примере команда MOVEW используется чтобы скопировать 16-битное слово из ячейки V2000 в V3000 когда бит C100 установлен (On).

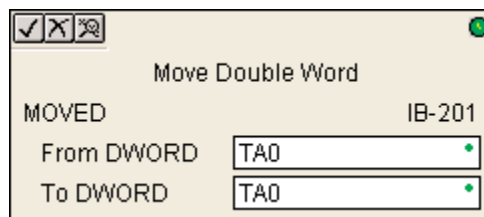


Move Double Word (MOVED) (IB-201)

Эта команда копирует ДВА слова данных из ячеек памяти в другие ячейки либо записывает шестнадцатеричную константу, заданную двойным словом, в ячейки памяти.

Параметры:

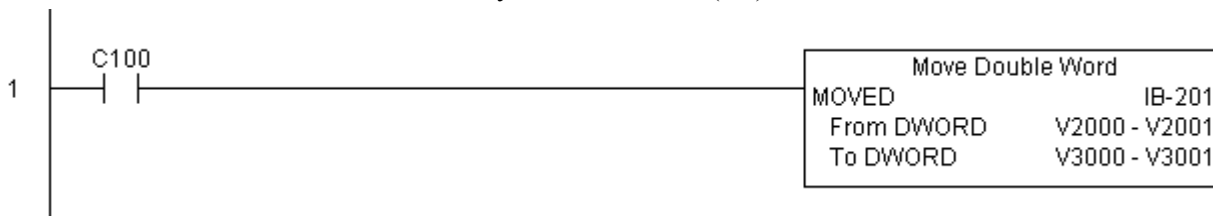
- **From DWORD:** адрес ячейки памяти или константа – источник данных (двойное слово).
- **To DWORD:** адрес ячейки памяти – приемник данных.



Параметр	Диапазон DL06
From DWORD V,P,K	Конст. K0-FFFFFFFF, См. карту памяти DL06: Слова данных
To DWORD..... V,P	См. карту памяти DL06: Слова данных

Пример:

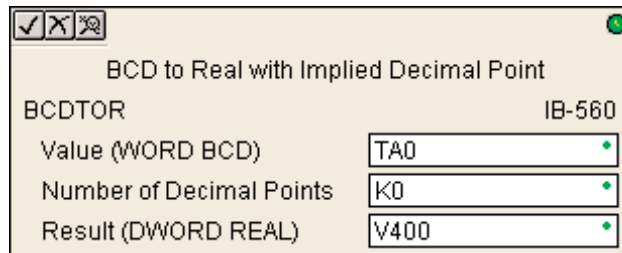
В этом примере команда MOVED используется чтобы скопировать 32-битное слово из ячейки V2000 и V2001 в V3000 и V3001 когда бит C100 устанавливается (On).



Математические команды

BCD to Real with Implied Decimal Point (BCDTOR) (IB-560)

Преобразует слово в формате BCD (4 знака) в вещественное число с дробной частью. Дробная часть задается параметром Number of Decimal Points, который указывает количество знаков после десятичной точки. Например, BCD число K1234 с Number of Decimal Points = K1 будет преобразовано в R123.4.



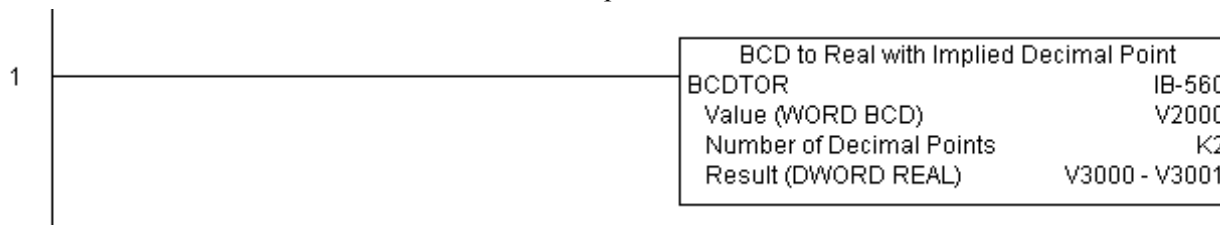
Параметры:

- **Value (WORD BCD):** адрес ячейки V-памяти или константа
- **Number of Decimal Points:** количество знаков после десятичной точки.
- **Result (DWORD REAL):** адрес ячейки V-памяти куда помещается результат преобразования

Параметр	Диапазон DL06
Value (WORD BCD) V,P,K	K0-9999 или См. карту памяти DL06 - Слова данных
Number of Decimal Points K	Константа K0-4
Result (DWORD REAL) V	См. карту памяти DL06 - Слова данных

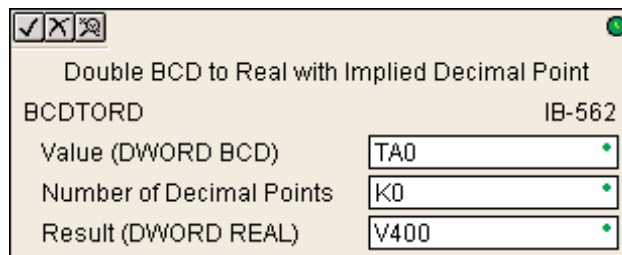
Пример:

В этом примере BCDTOR используется для преобразования 16-битного BCD числа в ячейке V2000 в вещественное 32-битное число с дробной частью. Результат сохраняется в ячейки V3000 и V3001. Константой K2 задано два знака после точки в дробной части.



Double BCD to Real with Implied Decimal Point (BCDTORD) (IB-562)

Преобразует двойное слово в формате BCD (8 знаков) в вещественное число с дробной частью. Дробная часть задается параметром Number of Decimal Points, который указывает количество знаков после десятичной точки. Например, BCD число K12345678 с Number of Decimal Points = K5 будет преобразовано в R123.45678.



Параметры:

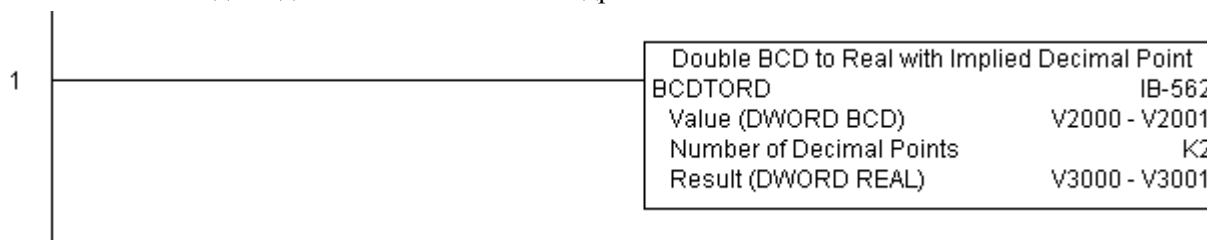
- **Value (DWORD BCD):** адрес ячейки V-памяти или константа
- **Number of Decimal Points:** количество знаков после десятичной точки.
- **Result (DWORD REAL):** адрес ячейки V-памяти куда помещается результат преобразования

Параметр	Диапазон DL06
Value (DWORD BCD) V,P,K	K0-99999999 или См. карту памяти DL06 - Слова данных
Number of Decimal Points K	Константа K0-8
Result (DWORD REAL) V	См. карту памяти DL06 - Слова данных

Пример:

В этом примере BCDTORD используется для преобразования 32-битного BCD числа в ячейке V2000 в вещественное 32-битное число с дробной частью. Результат сохраняется в ячейки V3000 и V3001.

Константой K2 задано два знака после точки в дробной части.



Math - BCD (MATHBCD) (IB-521)

Формат команды Math - BCD позволяет вставлять сложные математические выражения, как это делается в Excel, VB, C++, и выполнять вычисления до 4 вложенных уровней в скобках.

В дополнение к + - * / доступны также: модуль числа (% aka Remainder), побитовые операции (& -AND, | -OR, ^ -XOR), и некоторые BCD-функции: преобразование в BCD (BCD), преобразование в двоичное число (BIN), преобразование BCD числа в дополнительный код до 10 (BCDCPL), преобразование из Gray Code (GRAY), инвертирование битов (INV), преобразование числа BCD/HEX в команды управления семисегментным дисплеем (SEG).

*Пример: ((V2000 + V2001) / (V2003 - K100)) * GRAY(V3000 & K001F)*

Адресуемые в выражении ячейки V-памяти должны быть одним словом в формате BCD. Промежуточные результаты могут быть до 32 бит, но пока конечный результат укладывается в 16 битное BCD слово, вычисление правильное. Можно ссылаться также на ячейки V-памяти с данными в двоичном формате, используя при этом функцию преобразования ячейки (BCD), но не выражения.

BCD(V2000) + BCD(V3000) – правильно

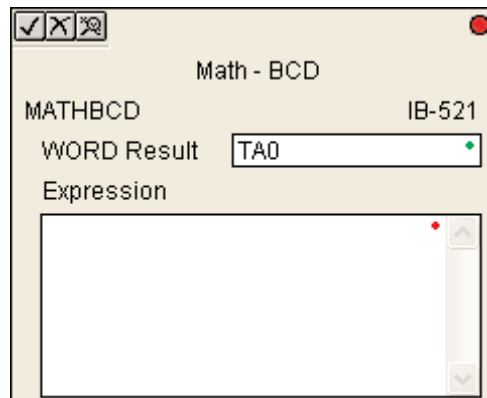
BCD(V2000 + V3000) – не правильно.

Результат выражения также в BCD формате. Для получения двоичного используйте функцию BIN ко всему выражению.

Параметры:

WORD Result: адрес ячейки V-памяти, куда будет помещен результат в формате BCD (результат должен быть в пределах 16 битового слова).

Expression: математическое выражение, которое будет вычислено и результат сохранен в ячейке WORD Result. Каждая ячейка, указываемая в выражении в качестве аргумента должна содержать данные в формате BCD.



Параметр	Диапазон DL06
WORD Result	См. карту памяти DL06 - Слова данных
Expression	Текст

Пример:

Здесь команда MATHBCD используется для вычисления математического выражения, в котором ячейка V1200 в формате BCD умножается на константу 1000, затем делится на 4095. Значение сохраняется в ячейке V2000 в формате BCD.



Math - Binary (MATHBIN) (IB-501)

Формат команды Math - Binary позволяет вставлять сложные математические выражения, как это делается в Excel, VB, C++, и выполнять вычисления до 4 вложенных уровней в скобках.

В дополнение к + - * / доступны также: модуль числа (% aka Remainder), сдвиг вправо (>>), сдвиг влево (<<), побитовые операции (& -AND, | -OR, ^ -XOR), и некоторые двоичные функции: преобразование в BCD (BCD), преобразование в двоичное число (BIN), декодирование битов (DECO), кодирование битов (ENCO), инвертирование битов (INV), преобразование числа HEX в команды управления семисегментным дисплеем (SEG), суммирование битов (SUM).

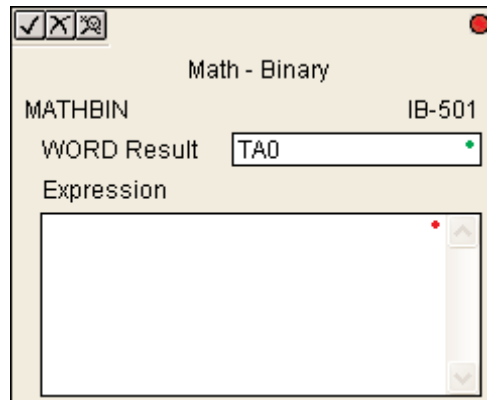
*Пример: ((V2000 + V2001) / (V2003 - K10)) * SUM(V3000 & K001F)*

Адресуемые в выражении ячейки V-памяти должны быть одним словом в двоичном формате. Промежуточные результаты могут быть до 32 бит, но пока конечный результат укладывается в 16 битное двоичное слово, вычисление правильное. Можно ссылаться также на ячейки V-памяти с данными в BCD формате, используя при этом функцию преобразования ячейки (BIN), но не выражения.

BIN(V2000) + BIN(V3000) – правильно

BIN(V2000 + V3000) – не правильно.

Результат выражения также в двоичном формате. Для получения BCD результата используйте функцию BCD ко всему выражению.



Параметры:

WORD Result: адрес ячейки V-памяти, куда будет помещен результат в двоичном формате (результат должен быть в пределах 16 битового слова).

Expression: математическое выражение, которое будет вычислено и результат сохранен в ячейке WORD Result. Каждая ячейка, помещаемая в выражение в качестве аргумента должна быть в двоичном формате.

Параметр	Диапазон DL06
WORD Result	См. карту памяти DL06 - Слова данных
Expression	Текст

Пример:

Здесь команда MATHBCD используется для вычисления математического выражения, в котором ячейка V1200 в двоичном формате умножается на константу 1000, затем делится на 4095. Значение сохраняется в ячейке V2000 в двоичном формате.



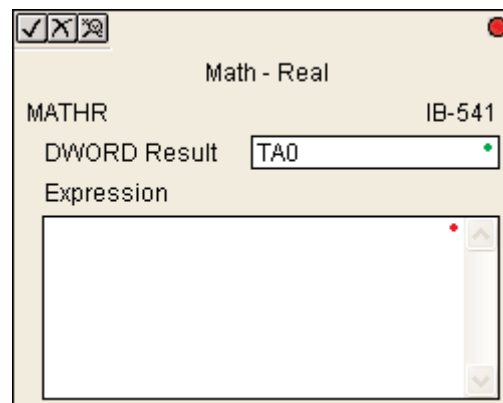
Math - Real (MATHR) (IB-541)

Формат команды Math - Real позволяет вставлять сложные математические выражения, как это делается в Excel, VB, C++, и выполнять вычисления до 4 вложенных уровней в скобках.

В дополнение к + - * / доступны также: побитовые операции (& -AND, | -OR, ^ -XOR), и некоторые функции: арккосинус (ACOSR), арксинус (ASINR), арктангенс (ATANR), косинус (COSR), конвертирование радиан в градусы (DEGR), инвертирование битов (INV), конвертирование градусов в радианы (RADR), преобразование числа HEX в команды управления семисегментным дисплеем (SEG), синус (SINR), квадратный корень (SQRT), тангенс (TANR).

*Пример: ((V2000 + V2002) / (V2004 - R2.5)) * SINR(RADR(V3000 / R10.0))*

Все адресуемые в выражении ячейки V-памяти должны быть ДВОЙНЫМ словом в вещественном формате.



Параметры:

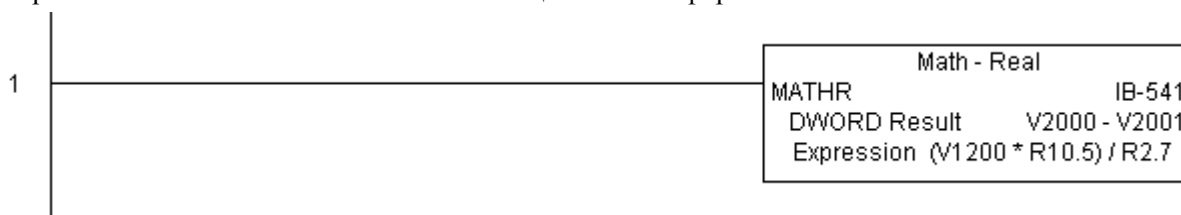
DWORD Result: адрес ячейки V-памяти, куда будет помещен результат – вещественное двойное слово.

Expression: математическое выражение, которое будет вычислено и результат сохранен в ячейке с адресом в параметре DWORD Result. Каждая ячейка, помещаемая в выражение в качестве аргумента должна быть в вещественном формате.

Параметр	Диапазон DL06
DWORD Result	См. карту памяти DL06 - Слова данных
Expression	Текст

Пример:

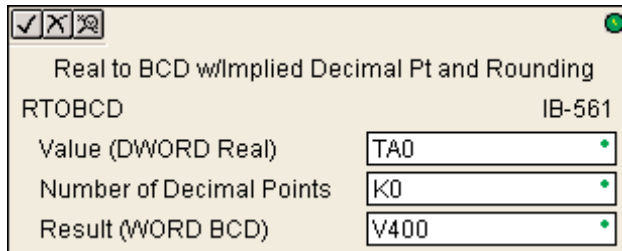
Здесь команда MATHR используется для вычисления математического выражения, в котором ячейка V1200 в вещественном формате умножается на константу 10.5, затем делится на 2.7. Значение сохраняется в ячейках V2000 и V2001 в вещественном формате.



Real to BCD with Implied Decimal Point and Rounding (RTOBCD) (IB-561)

Преобразует вещественное число в 4-значное BCD число, при этом определяя положение десятичной точки и выполняя округление. Положение точки определяется параметром Number of Decimal Points (K0-K4).

Например, вещественное число R56.74 с Number of Decimal Points = K1 будет преобразовано в BCD число 567. Если число знаков после точки (Number of Decimal Points) указать 0, то результат будет округлен до 57. Если вещественное число отрицательно, то результат будет равен положительному абсолютному значению.



Параметры:

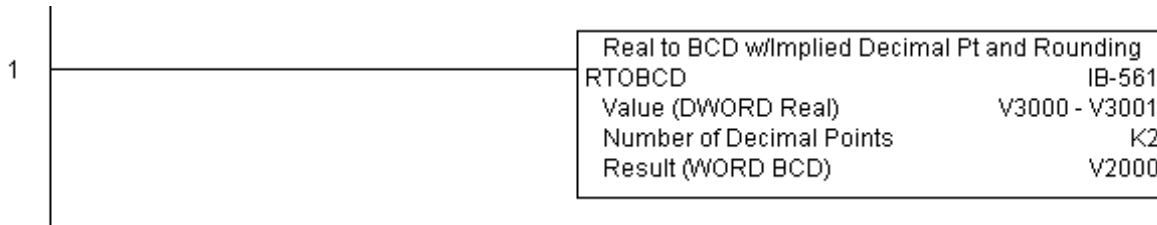
- **Value (DWORD Real):** адрес ячейки V-памяти или константа – вещественное двойное слово
- **Number of Decimal Points:** количество знаков после десятичной точки.
- **Result (WORD BCD):** адрес ячейки V-памяти куда помещается результат преобразования в BCD формате

Параметр	Диапазон DL06
Value (DWORD Real) V,P,R	R или См. карту памяти DL06 - Слова данных
Number of Decimal Points K	Константа K0-4
Result (WORD BCD) V	См. карту памяти DL06 - Слова данных

Пример:

В этом примере RTOBCD используется для преобразования 32-битного вещественного числа в ячейках V3000 и V3001 в 4-значное BCD число. Результат сохраняется в ячейке V2000.

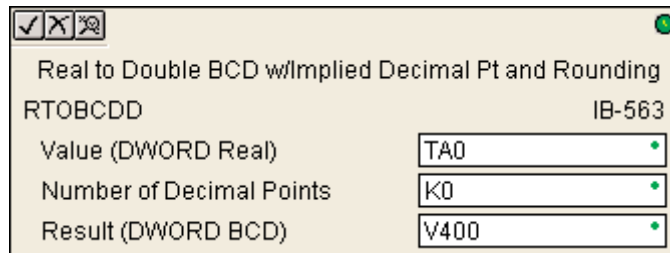
Константой K2 задано два знака после точки в дробной части.



Real to Double BCD with Implied Decimal Point and Rounding (RTOBCDD) (IB-563)

Преобразует абсолютное значение двойного вещественного числа в 8-значное BCD число, при этом определяя положение десятичной точки и выполняя округление. Положение точки определяется параметром Number of Decimal Points (K0-K8).

Например, вещественное число R38156.74 с Number of Decimal Points = K1 будет преобразовано в BCD число 381567. Если число знаков после точки (Number of Decimal Points) указать 0, то результат будет округлен до 38157. Если вещественное число отрицательно, то результат будет равен положительному абсолютному значению.



Параметры:

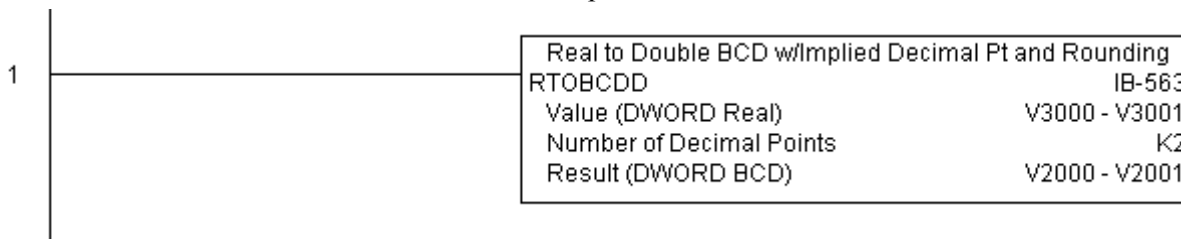
- **Value (DWORD Real):** адрес ячейки V-памяти или константа – вещественное двойное слово
- **Number of Decimal Points:** количество знаков после десятичной точки.
- **Result (DWORD BCD):** адрес ячейки V-памяти куда помещается результат преобразования в BCD формате

Параметр	Диапазон DL06
Value (DWORD Real) V,P,R	R или См. карту памяти DL06 - Слова данных
Number of Decimal Points K	Константа K0-8
Result (DWORD BCD) V	См. карту памяти DL06 - Слова данных

Пример:

В этом примере RTOBCDD используется для преобразования 32-битного вещественного числа в ячейках V3000 и V3001 в 8-значное BCD число. Результат сохраняется в ячейках V2000 и V2001.

Константой K2 задано два знака после точки в дробной части.



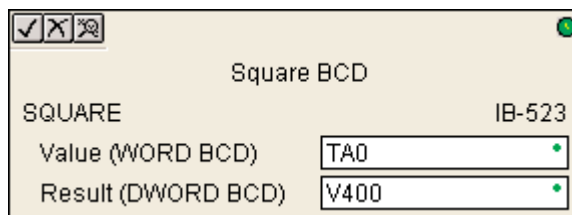
Square BCD (SQUARE) (IB-523)

Команда Square BCD возводит в квадрат слово в формате BCD и результат сохраняет в двойном слове в формате BCD.

Параметры:

Value (WORD BCD): адрес ячейки V-памяти (одно слово – 4 десятичных разряда) или константа (от 0 до 9999) для возведения в квадрат.

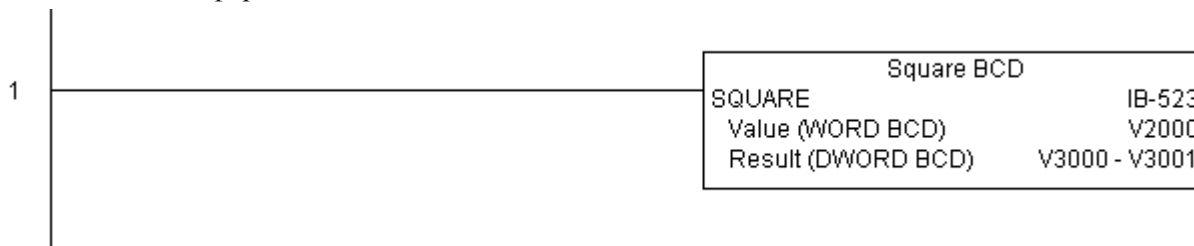
Result (DWORD BCD): адрес ячейки V-памяти, куда будет помещен результат вычисления – двойное слово в BCD формате.



Параметр	Диапазон DL06
Value (WORD BCD) V,P,K	См. карту памяти DL06 - Слова данных, Константы K0-K9999
Result (DWORD BCD) V	См. карту памяти DL06 - Слова данных

Пример:

Здесь команда MATHBCD используется для возведения в квадрат ячейки V2000 в BCD формате (4 десятичных разрядов). Результат сохраняется в виде двойного слова (8 десятичных разрядов) в ячейках V3000 и V3001 в формате BCD.



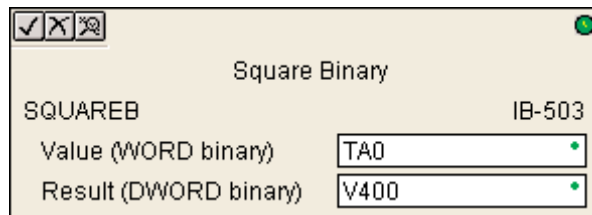
Square Binary (SQUAREB) (IB-503)

Команда Square Binary возводит в квадрат слово в двоичном формате и результат сохраняет в двойном слове в двоичном формате.

Параметры:

Value (WORD Binary): адрес ячейки V-памяти (одно слово – 4 десятичных разряда) или константа (от 0 до 9999) для возведения в квадрат.

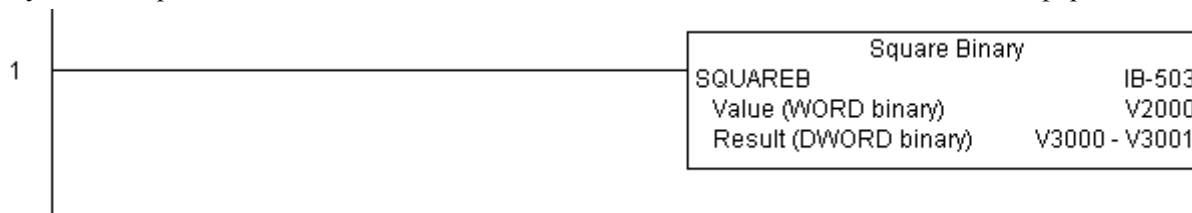
Result (DWORD Binary): адрес ячейки V-памяти, куда будет помещен результат вычисления – двойное слово в двоичном формате.



Параметр	Диапазон DL06
Value (WORD Binary) V,P,K	См. карту памяти DL06: Слова данных, Константы K0-K65535
Result (DWORD Binary) V	См. карту памяти DL06: Слова данных.

Пример:

Здесь команда SQUAREB используется для возведения в квадрат ячейки V2000 в двоичном формате. Результат сохраняется в виде двойного слова в ячейках V3000 и V3001 в двоичном формате.

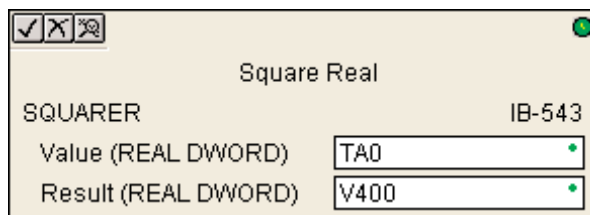


Square Real (SQUARER) (IB-543)

Команда Square Real возводит в квадрат ДВОЙНОЕ слово в вещественном формате, и результат сохраняет также в двойном слове в вещественном формате.

Параметры:

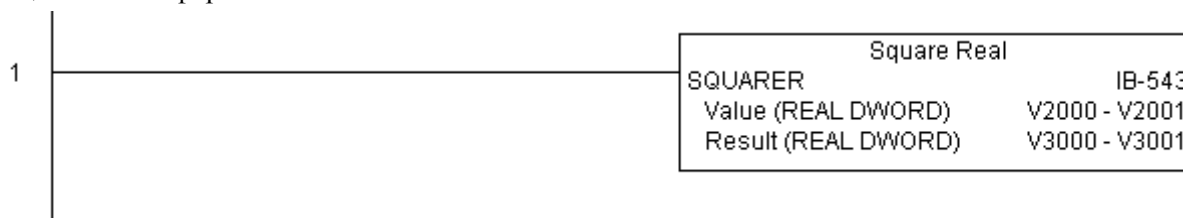
- **Value (REAL DWORD):** адрес ячейки V-памяти (двойное слово) или число для возведения в квадрат.
- **Result (REAL DWORD):** адрес ячейки V-памяти, куда будет помещен результат вычисления – двойное слово в вещественном формате.



Параметр	Диапазон DL06
Value (REAL DWORD) V,P,R	R или См. карту памяти DL06: Слова данных
Result (REAL DWORD) V	См. карту памяти DL06: Слова данных.

Пример:

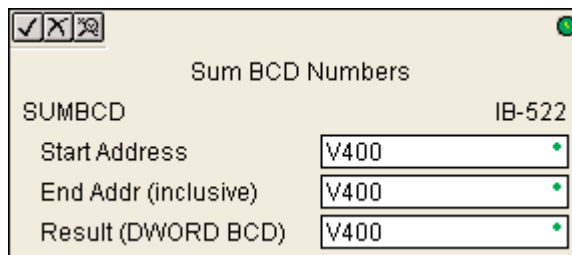
Здесь команда SQUARER используется для возведения в квадрат вещественного 32-битного числа в ячейках V2000 и V2001. Результат сохраняется в виде двойного слова в ячейках V3000 и V3001 в вещественном формате.



Sum BCD Numbers (SUMBCD) (IB-522)

Выполняет операцию сложения набора чисел в формате BCD-слово. Набор задается областью V-памяти, указывая адрес первого и последнего числа включительно. При активизации команды (можно использовать дифференциальный контакт) суммируются все числа в наборе.

SUMBCD может использоваться для вычисления среднего арифметического.



Параметры:

Start Address: начальный адрес памяти первого суммируемого числа в формате BCD.

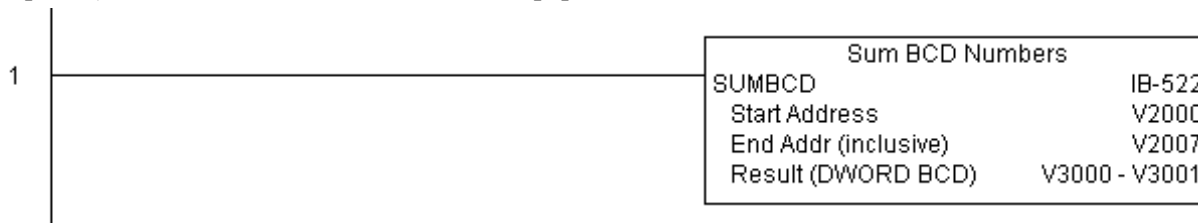
End Addr: адрес памяти последнего суммируемого числа в формате BCD.

Result: адрес памяти в который записывается результат сложения – двойное слово в формате BCD.

Параметр	Диапазон DL06
Start Address V	См. карту памяти DL06 - Слова данных
End Address (inclusive) V	См. карту памяти DL06 - Слова данных
Result (DWORD BCD) V	См. карту памяти DL06 - Слова данных.

Пример:

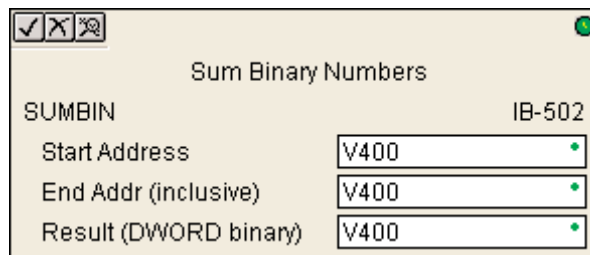
Здесь команда SUMBCD используется для сложения всех ячеек памяти с V2000 по V2007 заданных в BCD формате (4 десятичных разрядов). Результат сохраняется в виде двойного слова (8 десятичных разрядов) в ячейках V3000 и V3001 в BCD формате.



Sum Binary Numbers (SUMBIN) (IB-502)

Выполняет операцию сложения набора чисел в формате двоичного слова (16 бит). Набор задается областью V-памяти, указывая адрес первого и последнего числа включительно. При активизации команды (можно использовать дифференциальный контакт) суммируются все числа в наборе.

SUMBIN может использоваться для вычисления среднего арифметического.



Параметры:

Start Address: начальный адрес памяти первого суммируемого двоичного числа.

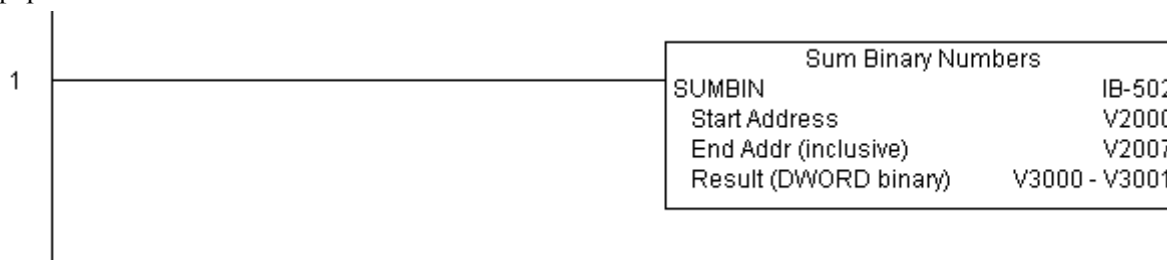
End Addr: адрес памяти последнего суммируемого двоичного числа.

Result: адрес памяти в который записывается результат сложения – двойное двоичное слово.

Параметр	Диапазон DL06
Start Address V	См. карту памяти DL06 - Слова данных
End Address (inclusive) V	См. карту памяти DL06 - Слова данных
Result (DWORD Binary) V	См. карту памяти DL06 - Слова данных

Пример:

В этом примере команда SUMBIN используется для сложения всех ячеек памяти с V2000 по V2007 в двоичном формате. Результат сохраняется в виде двойного слова в ячейках V3000 и V3001 в двоичном формате.



Sum Real Numbers (SUMR) (IB-542)

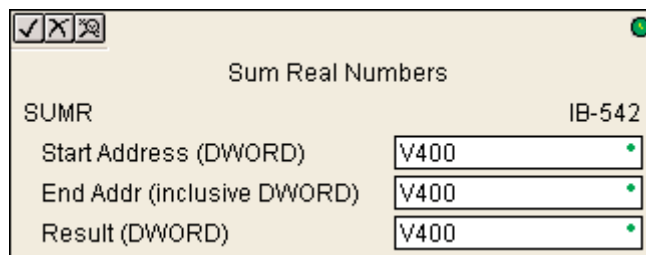
Выполняет операцию сложения набора вещественных чисел в формате ДВОЙНОГО слова. Набор задается областью V-памяти, указывая адрес первого и последнего числа включительно.

Помните, что вещественные числа занимают по два слова V-памяти каждое, это значит количество суммируемых чисел в два раза меньше длины указываемой области памяти. В End Addr можно указывать адрес любого – младшего и старшего слова.

Например, если вы хотите сложить 4 вещественных числа в ячейках с V2000 по V2007, то в End Addr можно указать V2006 или V2007.

При активизации команды (можно использовать дифференциальный контакт) суммируются все числа в наборе.

SUMR может использоваться для вычисления среднего арифметического.



Параметры:

Start Address (DWORD): начальный адрес памяти первого суммируемого числа.

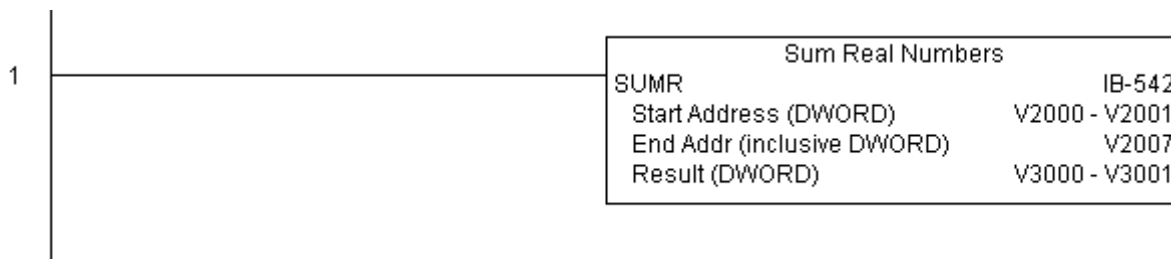
End Addr (DWORD): адрес памяти последнего суммируемого числа.

Result (DWORD): адрес памяти в который записывается результат сложения – двойное вещественное слово.

Параметр	Диапазон DL06
Start Address (inclusive DWORD) V	См. карту памяти DL06 - Слова данных
End Address (inclusive DWORD) V	См. карту памяти DL06 - Слова данных
Result (DWORD) V	См. карту памяти DL06 - Слова данных

Пример:

В этом примере команда SUMR используется для сложения всех чисел с V2000 по V2007 в формате вещественного двойного слова. Результат сохраняется в виде вещественного двойного слова (32 бита) в ячейках V3000 и V3001.



Коммуникационные команды

ECOM100 Configuration (ECOM100) (IB-710)

Эта команда выполняет настройку одного ECOM100 модуля, с которым работают коммуникационные инструкции: ECRX - ECOM100, Network Read, ECEMAIL - ECOM100, Send EMail, ECIPSUP - ECOM100 IP Setup, и пр. Поэтому эта команда должна быть в самом начале программы. Параметр MSG Buffer задает начальный адрес области памяти длиной 65 слов для буфера, то есть он занимает 101 восьмеричный адрес (например, с V1400 по V1500).

Для каждого коммуникационного модуля ECOM100 должен быть своя отдельная команда настройки. Все ячейки памяти, указываемые в этой команде, предназначаются только для этого настройки модуля, и не должны использоваться далее в программе.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

Для большинства функций ECOM100 микрпереключатель 7 на плате модуля должен быть включен (On). Для использования только функций Network Read and Write IBoxes (ECRX, ECWX) он может быть выключен (Off).

ECOM100 Config	
ECOM100	IB-710
ECOM100 #	K0
Slot	K1
Status	V400
Workspace	V400
Msg Buffer (65 WORDs)	V400

Параметры:

- **ECOM100#:** логический номер модуля, присваиваемый конкретному модулю ECOM100. Команды работающие с этим модулем должны адресоваться на этот номер.
- **Slot:** номер слота, в котором установлен модуль.
- **Status:** адрес ячейки памяти, используемый для регистра статуса модуля.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Msg Buffer:** начальный адрес области памяти для конфигурации. Длина области 65 слов.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-K255
Slot K	Константа: K1-4
Status V	См. карту памяти DL06 - Слова данных
Workspace V	См. карту памяти DL06 - Слова данных
Msg Buffer (65 words used) V	См. карту памяти DL06 - Слова данных

Пример:

Команда ECOM100 Config координирует взаимодействия между всеми другими коммуникационными инструкциями одного модуля. Эта команда должна быть первой из них, в самом начале программы.

В этом примере задается логический номер модуля в слоте 3 как №0 . Остальные команды работы с этим модулем должны указывать в своем параметре ECOM100# тоже K0.

Регистр статуса Status сообщает о завершении либо ошибке передачи данных. Регистр Workspace хранит состояние модуля для обеспечения взаимодействия других команд работающих с модулем ECOM100. Буфер сообщений используется другими командами для формирования отправляемого сообщения и приема. Адреса этих регистров не должны быть использованы где-либо еще в программе.

1

ECOM100 Config		
ECOM100		IB-710
ECOM100 #		K0
Slot		K3
Status		V1501
Workspace		V1502
Msg Buffer (65 WORDs)		V1400 - V1500

ECOM100 Disable DHCP (ECDHCPD) (IB-736)

Эта команда отключает динамическую адресацию модуля (DHCP), позволяя задать статический IP адрес модуля. Для настройки TCP/IP используйте утилиту NetEdit3, иначе это можно сделать командой ECOM100 IP Setup (ECIPUP) из программы контроллера, либо отдельными командами: ECOM Write IP Address (ECWRIP), ECOM Write Gateway Address (ECWRGWA), and ECOM100 Write Subnet Mask (ECWRSNM).

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно. В случае ошибки параметр Error Code получит код ошибки модуля (меньше 100) или программы контроллера (больше 1000).

Настройка модуля «Отключить DHCP» сохраняется во Flash ROM памяти модуля и при выполнении этой команды модуль в момент записи остается недоступным не менее половины секунды. Тем не менее, крайне рекомендуется выполнять эту команду только один раз в первом скан-цикле программы. Эта команда срабатывает по переднему фронту, поэтому используйте нормально закрытое реле SP0 (STR NOT First Scan) вместе с командой в программе.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).

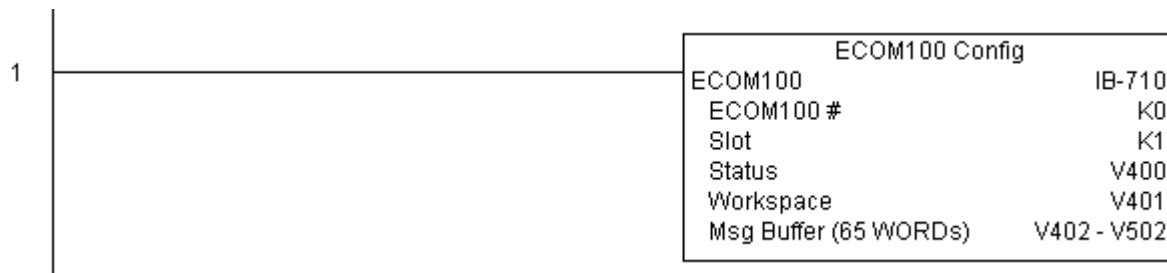
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ESxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Error Code**: адрес ячейки памяти, куда сохраняется код ошибки при неудачном запросе.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-K255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Error Code V	См. карту памяти DL06 - Слова данных

Пример:

Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций одного модуля ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер. Если необходимо переместить модуль в другой слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля, а Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100. Ячейка V401 используется для координации логики в остальных инструкциях, которые работают с модулем.



Цепь 2: На втором скан-цикле программы отключается DHCP. DHCP – это протокол, с помощью которого подключенный к модулю компьютер автоматически назначает ему: IP адрес, шлюз, маску подсети. Обычно отключение DHCP выполняется назначением статического IP адреса либо с помощью утилиты NetEdit, либо в инструкции ECOM100 IP Setup IBoxes, а команда ECOM100 Disable DHCP позволяет сделать это в программе контроллера. Эта команда срабатывает только в момент изменения логического условия исполнения команды ECDHCPD с Off на On. При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101. При ошибке её код записывается в ячейку V2000.



ECOM100 Enable DHCP (ECDHCPE) (IB-735)

Эта команда настраивает интерфейсный модуль ECOM100 на получение настроек TCP/IP от сервера DHCP (IP адрес, шлюз, маску). Команда выполняется по переднему фронту срабатывания входного условия.

Команда успешно завершается как только модуль получает от сервера настройки TCP/IP. Поскольку сервер может быть недоступен, то завершение команды определяется параметром Timeout; при этом устанавливается признак (Error) и код ошибки Error Code = 1004 (десятич.).

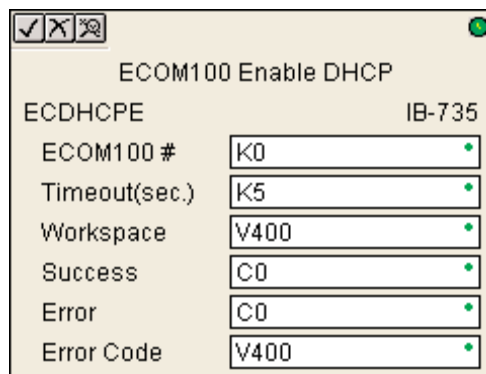
Для прямой установки фиксированных статических TCP/IP настроек используйте команду ECOM100 IP Setup (ECIPSUP) IBox 717.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно. В случае ошибки параметр Error Code получит код ошибки модуля (меньше 100) или программы контроллера (больше 1000).

Настройка модуля «Включить DHCP» сохраняется во Flash ROM памяти модуля и при выполнении этой команды модуль в момент записи остается недоступным не менее половины секунды. Тем не менее, крайне рекомендуется выполнять эту команду только один раз в первом скан-цикле программы. Эта команда срабатывает по переднему фронту, поэтому используйте нормально закрытое реле SP0 (STR NOT First Scan) вместе с командой в программе.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).



Параметры:

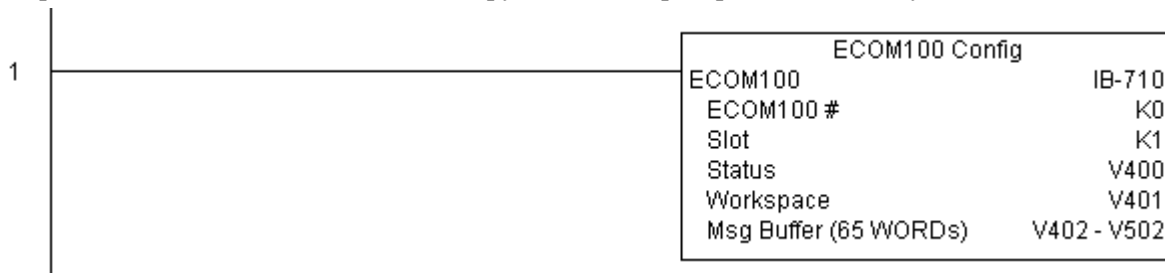
- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ESxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Timeout(sec)**: задает ожидаемое время в секундах для выполнения этой команды.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Error Code**: адрес ячейки памяти, куда сохраняется код ошибки при неудачном запросе.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Timeout (sec) K	Константа: K5-127
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Error Code V	См. карту памяти DL06 - Слова данных

Пример:

Цель 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций одного модуля ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер. Если необходимо переместить модуль в другой слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля, а Область памяти V402-V502 – общий буфер длиной 130

байт для инструкций, работающих с данным модулем ECOM100. Ячейка V401 используется для координации логики в остальных инструкциях, которые работают с модулем.



Цепь 2: На втором скан-цикле программы включается DHCP. DHCP – это протокол, с помощью которого подключенный к модулю компьютер автоматически назначает ему: IP адрес, шлюз, маску подсети. Обычно включение DHCP выполняется с помощью утилиты NetEdit, либо в инструкции ECOM100 IP Setup IBoxes, а команда ECOM100 Enable DHCP позволяет сделать это в программе контроллера. Эта команда срабатывает только в момент срабатывания логического условия на входе команды ECDHCPE с Off на On. По этой команде не просто выставляется бит «Включить DHCP» в модуле, но еще настраивается ежесекундная проверка: найден ли сервер DHCP и получен ли правильный IP адрес. Параметр Timeout также необходим. В случае если сервер не будет найден за указанное им время будет выставлен бит Error и код ошибки будет равен 1005 (десятич.). При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101. При ошибке её код записывается в ячейку V2000.



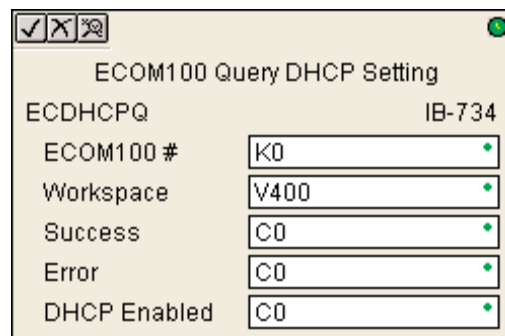
ECOM100 Query DHCP Setting (ECDHCPQ) (IB-734)

По переднему фронту логического условия эта команда определяет включен ли DHCP режим в модуле ECOM100. Если DHCP включен, то выставляется бит заданный параметром DHCP Enabled, иначе сбрасывается.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).



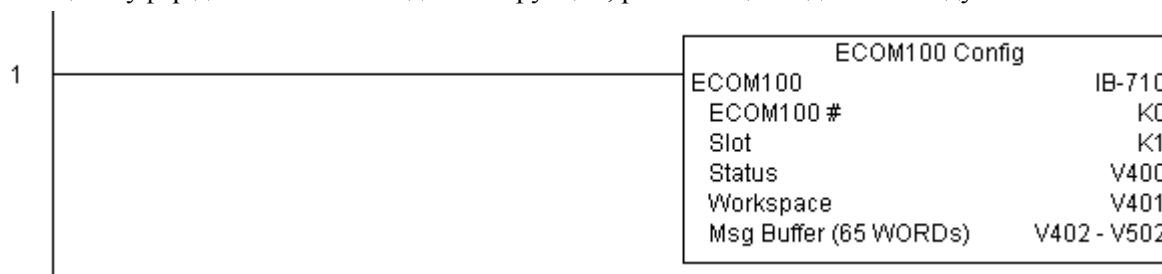
Параметры:

- **ECOM100#:** логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ЕСxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Success:** адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error:** адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **DHCP Enabled:** адрес бита, который будет показывать включен ли DHCP. После выполнения команды совместно с битом DHCP Enabled необходимо также проверять результат запроса в битах Success и Error.

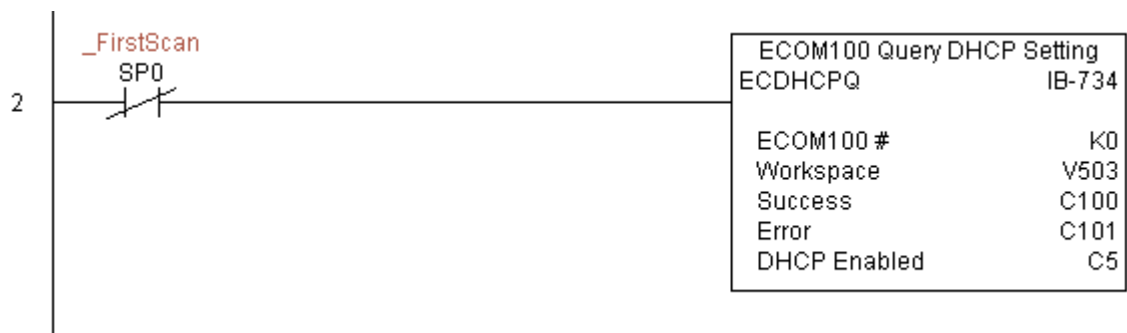
Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
DHCP Enabled X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля, а Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: На втором скан-цикле программы выполняется проверка настройки DHCP и результат сохраняется в бите C5. DHCP – это протокол, с помощью которого подключенный к модулю компьютер автоматически назначает ему: IP адрес, шлюз, маску подсети. Эта команда срабатывает только в момент срабатывания логического условия на входе команды ECDHCPQ с Off на On (по переднему фронту). При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101.



ECOM100 Send E-mail (ECEMAIL) (IB-711)

Эта инструкция работает также как E-mail клиент, посылая SMTP запросы на сервер SMTP для отправки E-mail сообщения по адресу, указанному в поле To, а также копии по списку адресов сохраненных в памяти модуля (для поля Cc). Инструкция запросы отправляет модулю с номером, заданным полем ECOM100#, который должен быть настроен командой ECOM100 Configuration в вашей программе выше.

В поле Body такие же как и в инструкциях PRINT и VPRINT возможности динамического добавления текста или значений переменных, а также вставки данных реального времени ("V2000 = " V2000:B).

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно. В случае ошибки параметр Error Code получит код ошибки модуля (меньше 100), либо протокола SMTP (между 100 и 999), или программы контроллера (больше 1000).

Поскольку модуль ECOM100 работает только как Email-клиент и требует доступ к SMTP-серверу, то должны быть правильно настроены SMTP параметры модуля через Web-страничку модуля или с помощью инструкции EMail Setup (ECEMSUP). Чтобы зайти на страничку модуля используйте любой Web-браузер и укажите в нем заранее заданный адрес модуля (например такой: http://192.168.12.86). Адрес задается командами ECOM100 Configuration (ECOM100) или Enable DHCP (ECDHCPE).

Длина E-mail сообщения ограничена примерно 100 символами/байтами, включая поля To, Subject, Body. Модуль ECOM100 позволяет заранее сохранить в своей памяти адреса для рассылки копий (для поля Cc), поэтому вы можете меньше адресов вписывать в поле To или даже вообще оставить пустым.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).

Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Error Code**: адрес ячейки памяти, куда сохраняется код ошибки при неудачном запросе.
- **To**: E-Mail адрес на который будет отправлено сообщение.
- **Subject**: тема сообщения.
- **Body**: поддерживает такие же как и в инструкциях PRINT и VPRINT возможности динамического добавления текста и значений переменных, а также вставки данных реального времени.

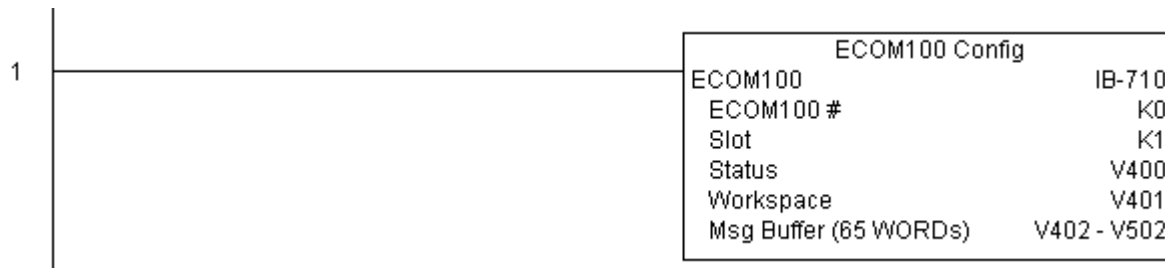
Параметр	Диапазон DL06
ECOM100#	Константа: K0-255
Workspace	См. карту памяти DL06 - Слова данных
Success	См. карту памяти DL06
Error	См. карту памяти DL06
Error Code	См. карту памяти DL06 - Слова данных

DirectSOFT5 Команды контроллера DL06

To:	Текст
Subject:	Текст
Body:	См. инструкции PRINT и VPRINT

Пример:

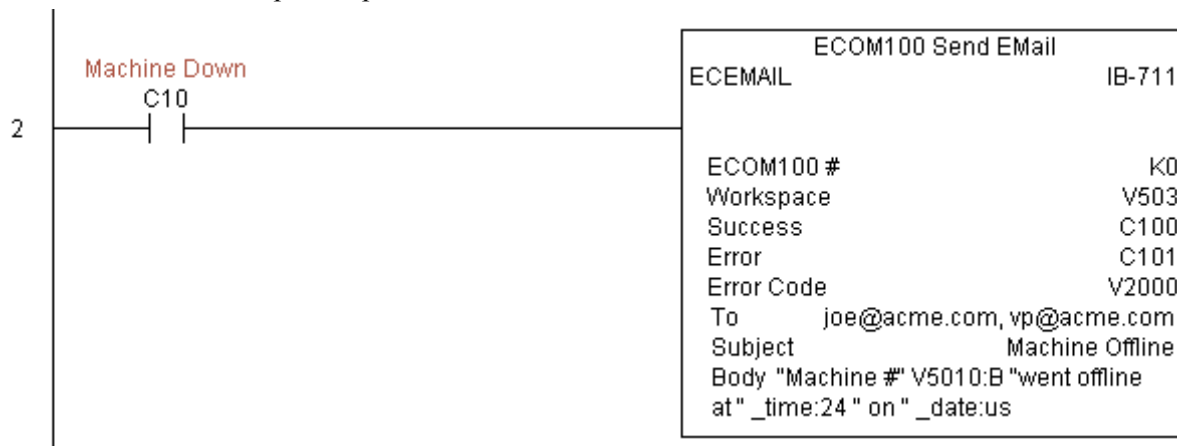
Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в каркасе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля, а Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: Когда машина останавливается, посылается сообщение адресату Joe для ее обслуживания и адресату VP для контроля производства, с указанием даты и времени остановки.

Эта команда выполняется только в момент срабатывания логического условия на входе команды ECEMAIL с Off на On (по переднему фронту). Это помогает предотвратить самопроизвольную рассылку.

Если сообщение опрaвлено успешно устанавливается бит C100, в противном случае – бит C101. Код ошибки может быть просмотрен в ячейке V2000.



ECOM100 Restore Default E-mail Setup (ECEMRDS) (IB-713)

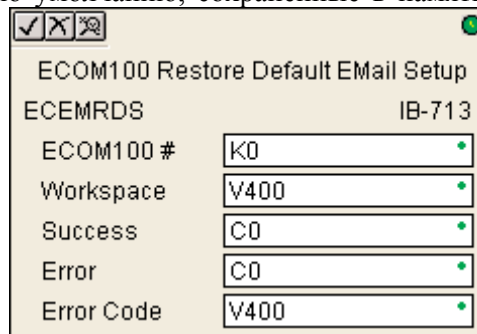
Эта инструкция восстанавливает Email оригинальные настройки по умолчанию, сохраненные в памяти модуля. Работает по переднему фронту.

Когда модуль ECOM100 включается в первый раз, он копирует настройки Email, сохраненные в ПЗУ, в рабочую память ОЗУ. Настройки в рабочей памяти можно поменять впоследствии из своей программы, используя команду ECOM100 EMail Setup (ECEMSUP). После модификации рабочей копии, позже, вы снова сможете восстановить оригинальные настройки с помощью команды ECOM100 Restore Default E-mail Setup (ECEMRDS).

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно. В случае ошибки параметр Error Code получит код ошибки модуля (меньше 100) или программы контроллера (больше 1000).

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).



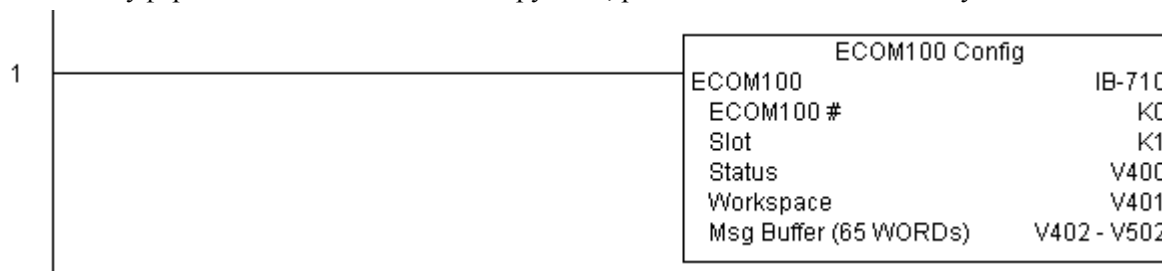
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Error Code**: адрес ячейки памяти, куда сохраняется код ошибки при неудачном запросе.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Error Code V	См. карту памяти DL06 - Слова данных

Пример:

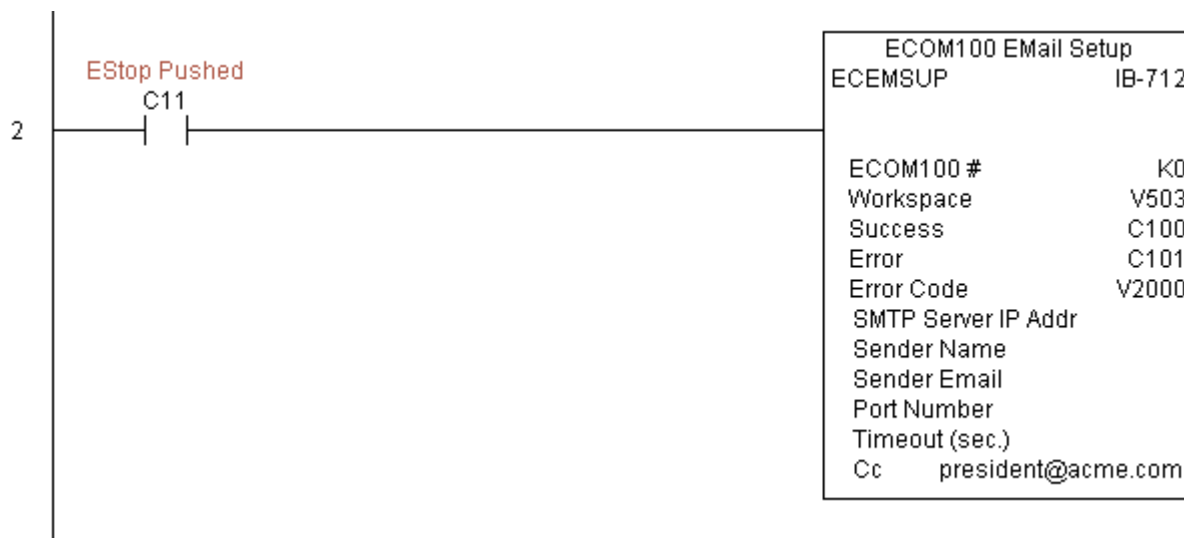
Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля, а Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: Когда сработает условие EStop, удостоверьтесь, что президент компании тоже получил все копии сообщений, которые были посланы.

DirectSOFT5 Команды контроллера DL06

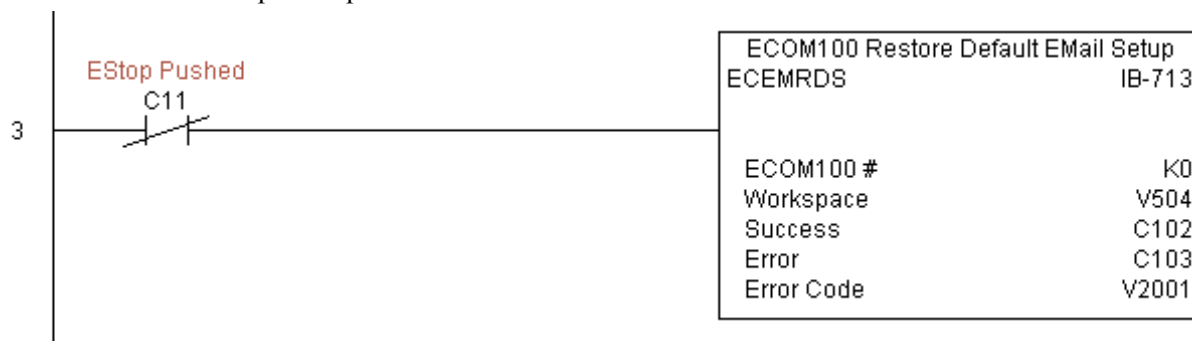
Команда ECOM100 EMail Setup IBox позволяет установить или изменить настройки SMTP Email сохраненные в памяти модуля ECOM100.



Цепь 3: Как только условие EStop станет ложным, удалите президента из списка рассылки (поле Cc) восстановив оригинальные настройки из памяти модуля.

Команда ECEMRDS выполняется по переднему фронту входного условия. Хранящаяся в ПЗУ модуля конфигурация Email будет скопирована поверх рабочей копии настроек в ОЗУ как только EStop станет On.

При успешном выполнении команды устанавливается бит C102, в противном случае – бит C103. Код ошибки может быть просмотрен в ячейке V2001.



ECOM100 E-mail Setup (ECEMSUP) (IB-712)

По этой команде модифицируются рабочие Email настройки в ОЗУ. Вы можете выбрать все или только некоторые поля для изменения этой командой.

Обратите внимание, что эти изменения кумулятивные, то есть если вы выполняете несколько таких команд последовательно, то изменения будут производиться каждой командой в порядке их выполнения. Кроме того, вы можете восстановить в рабочую копию настройки, сохраненные в ПЗУ модуля, используя команду ECOM100 Restore Default EMail Setup (ECEMRDS).

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно. В случае ошибки параметр Error Code получит код ошибки модуля (меньше 100) или программы контроллера (больше 1000).

Имеется ограничение примерно в 100 символов/байт для общей длины настроечных данных в одной инструкции. Но, можно последовательно использовать несколько таких команд. Так, некоторые поля могут быть модифицированы одной командой ECEMSUP, другие следующими. Используйте, например, настройку списка рассылки копий (поле Cc) в отдельной команде ECEMSUP.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).

Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ESxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Error Code**: адрес ячейки памяти, куда сохраняется код ошибки при неудачном запросе.
- **SMTP Server IP Addr**: необязательный параметр, задающий IP адрес SMTP сервера для отправки сообщений.
- **Sender Name**: необязательный параметр, определяющий имя отправителя, которое появляется в поле From сообщения у тех, кто его получает.
- **Sender Email**: необязательный параметр, задающий Email адрес отправителя, который появляется в поле From сообщения у тех, кто его получает.
- **Port Number**: необязательный параметр, который указывает номер порта TCP/IP для отправки SMTP запросов серверу. Обычно его не меняют, спросите своего сетевого администратора если считаете, что он может быть другим.
- **Timeout (sec)**: необязательный параметр, задающий время ожидания (в секундах) отправки сервером сообщения всем адресатам.
- **Cc**: необязательный параметр, который создает список получателей копии сообщения. В нем указываются Email адреса.

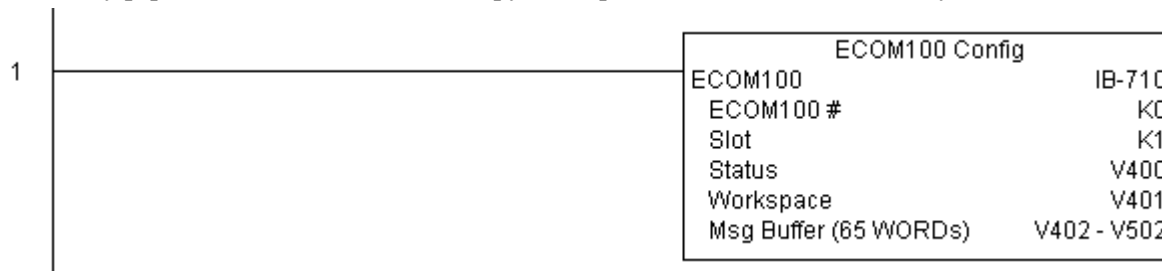
Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных

DirectSOFT5 Команды контроллера DL06

Success	X,Y,C,GX,GY,B	См. карту памяти DL06
Error	X,Y,C,GX,GY,B	См. карту памяти DL06
Error Code	V	См. карту памяти DL06 - Слова данных

Пример:

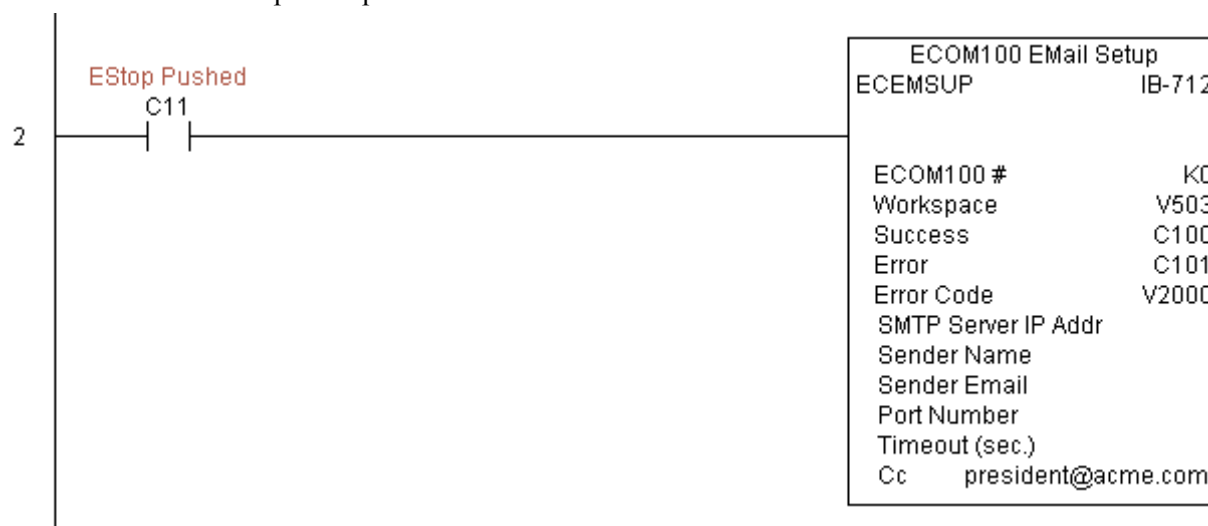
Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в каркасе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: Когда сработает условие EStop, удостоверьтесь, что президент компании тоже получил все копии сообщений, которые были посланы.

Команда ECOM100 EMail Setup IBox позволяет установить или изменить настройки SMTP Email сохраненные в памяти модуля ECOM100. Она срабатывает по переднему фронту входного условия. При включении модуля Email настройки, сохраненные в ПЗУ модуля, копируются в их рабочую копию в ОЗУ. Вы можете менять их в рабочей копии, используя команду ECEMSUP. Для восстановления оригинальных настроек используйте команду Restore Default EMail Setup (ECEMRDS).

При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101. Код ошибки может быть просмотрен в ячейке V2000.



Цепь 3: Как только условие EStop станет ложным, удалите президента из списка рассылки (поле Cc) восстановив оригинальные настройки из памяти модуля.



ECOM100 IP Setup (ECIPSUP) (IB-717)

С помощью этой инструкции конфигурируются три TCP/IP параметра модуля ECOM100: IP адрес, маска подсети, адрес шлюза. Выполняется она по переднему фронту входного условия.

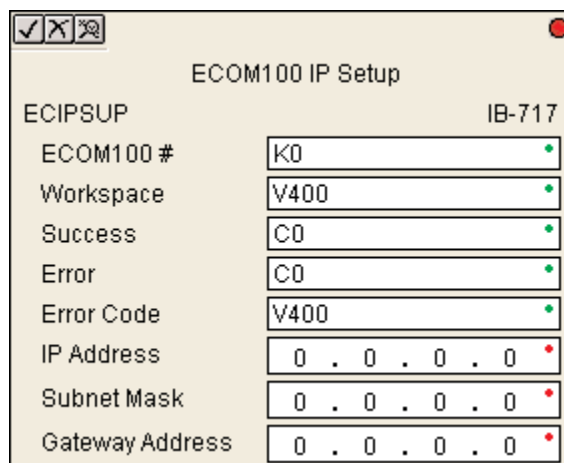
Номер модуля указывается в параметре ECOM100# в соответствии с физическим модулем настроенным предыдущей командой ECOM100 Configuration (ECOM100) по программе выше.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно. В случае ошибки параметр Error Code получит код ошибки модуля (меньше 100) или программы контроллера (больше 1000).

Эти настройки сохраняются во Flash ROM памяти модуля и при выполнении этой команды модуль в момент записи остается недоступным не менее половины секунды. Тем не менее, крайне рекомендуется выполнять эту команду только один раз в первом скан-цикле программы. Эта команда срабатывает по переднему фронту, поэтому вместе с командой в программе используйте нормально закрытое реле SP0 (STR NOT First Scan).

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).



Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Error Code**: адрес ячейки памяти, куда сохраняется код ошибки при неудачном запросе.
- **IP Address**: IP адрес модуля.
- **Subnet Mask**: маска подсети.
- **Gateway Address**: IP адрес шлюза.

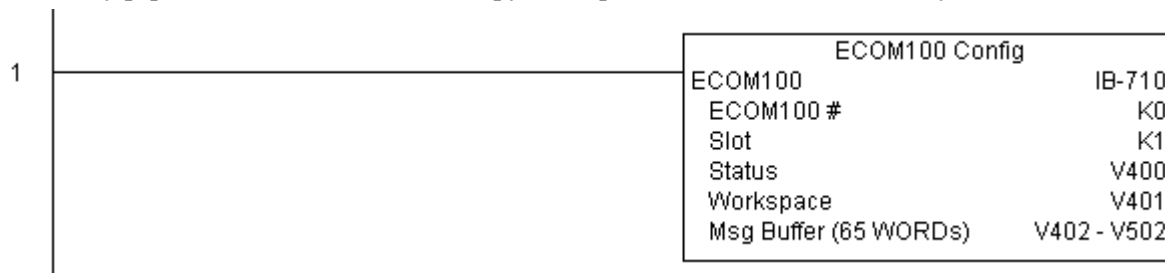
Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Error Code V	См. карту памяти DL06 - Слова данных
IP Address IP Address	0.0.0.1 – 255.255.255.254
Subnet Mask Address IP Address Mask	0.0.0.1 – 255.255.255.254
Gateway Address IP Address	0.0.0.1 – 255.255.255.254

Пример:

Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой

DirectSOFT5 Команды контроллера DL06

слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: На втором скан-цикле конфигурируются все TCP/IP параметры модуля:

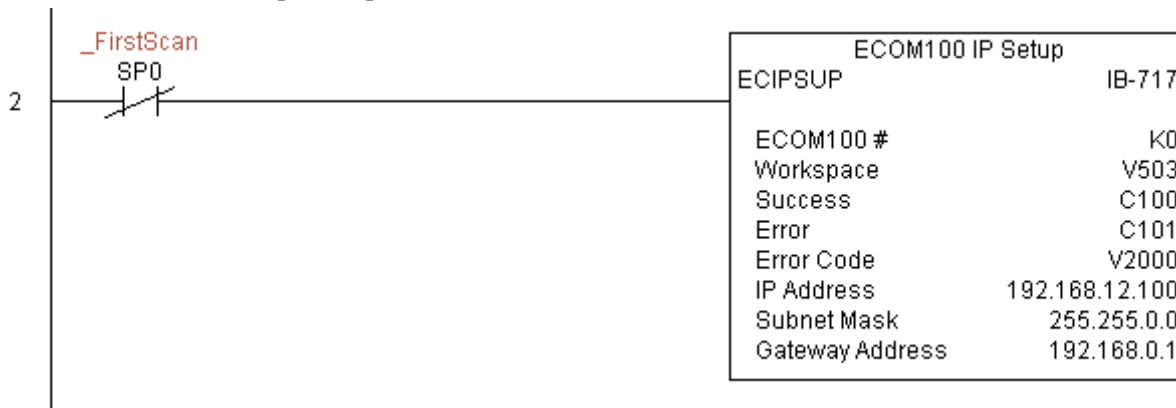
IP адрес: 192.168.12.100

Маска подсети: 255.255.0.0

Адрес шлюза: 192.168.0.1

Команда ECIPSUP выполняется по переднему фронту входного условия, в этот момент выполняется запись параметров TCP/IP в модуль ECOM100.

При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101. Код ошибки может быть просмотрен в ячейке V2000.



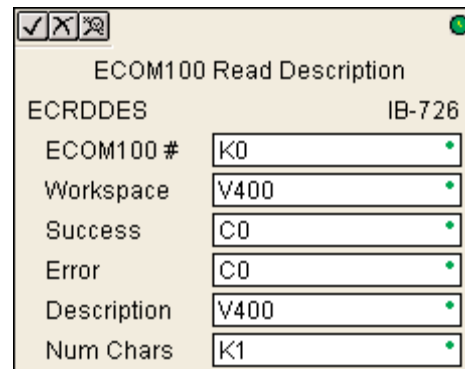
ECOM100 Read Description (ECRDDES) (IB-726)

С помощью этой команды можно получить описание установленного модуля ECOM100. Команда выполняется по переднему фронту.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).



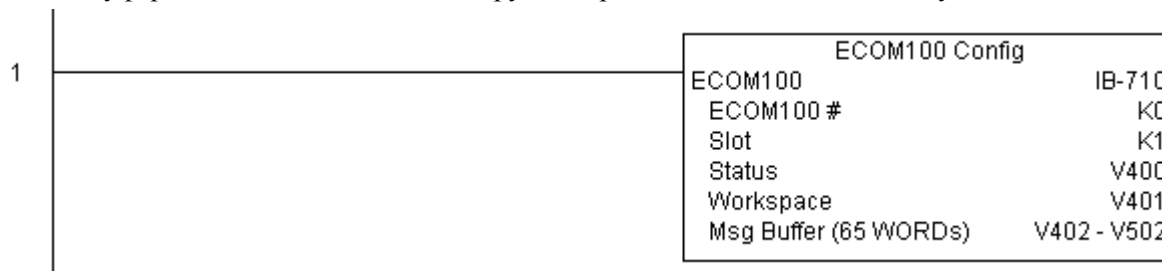
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Description**: адрес начальной ячейки блока памяти, в который будет помещена строка полученного описания модуля.
- **Num Char**: длина строки в символах (байтах) для чтения.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Description V	См. карту памяти DL06 - Слова данных
Num Chars K	K1-128

Пример:

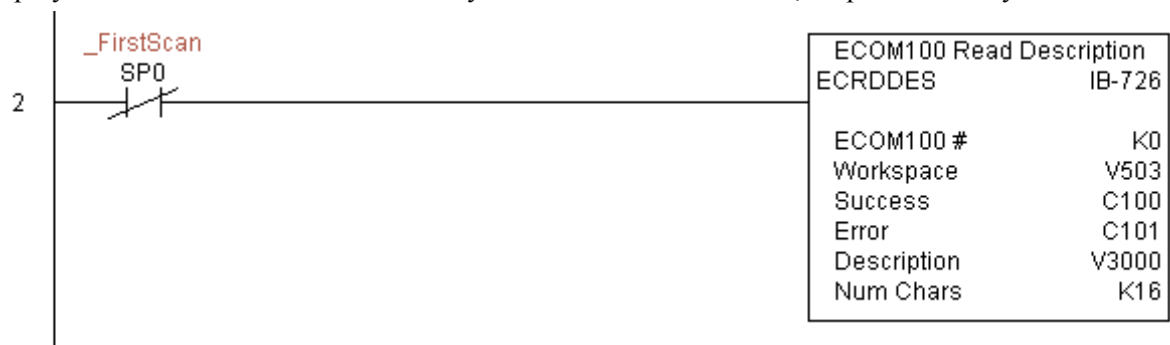
Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в каркасе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: На втором скан-цикле читается описание модуля ECOM100 и сохраняется в памяти по адресам с V3000 по V3007 (16 символов). Этот текст может быть выведен на экран монитора.

Команда ECRDDES выполняется по переднему фронту входного условия, в этот момент выполняется чтение строки описания.

При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101.



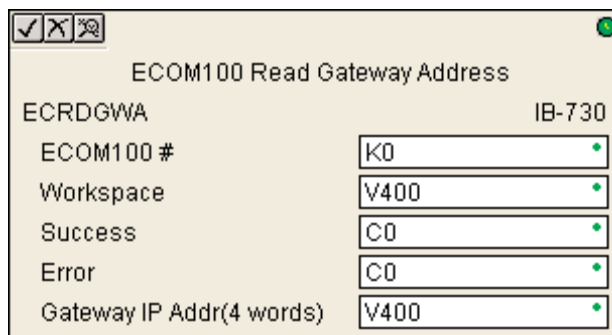
ECOM100 Read Gateway Address (ECRDGWA) (IB-730)

Эта инструкция читает настроенный IP адрес шлюза из модуля ECOM100 и сохраняет в четырех последовательных ячейках памяти в десятичном формате. Инструкция работает по переднему фронту.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).



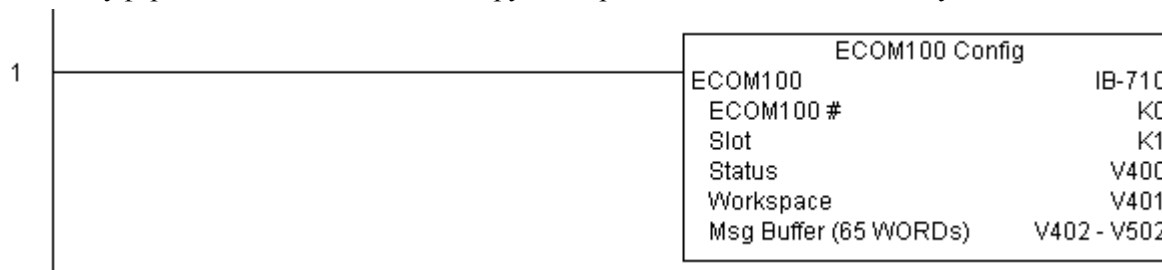
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Gateway IP Addr**: адрес первой ячейки памяти для сохранения адреса шлюза в четырех последовательных ячейках памяти в десятичном формате.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Gateway IP Address (4 Words) V	См. карту памяти DL06 - Слова данных

Пример:

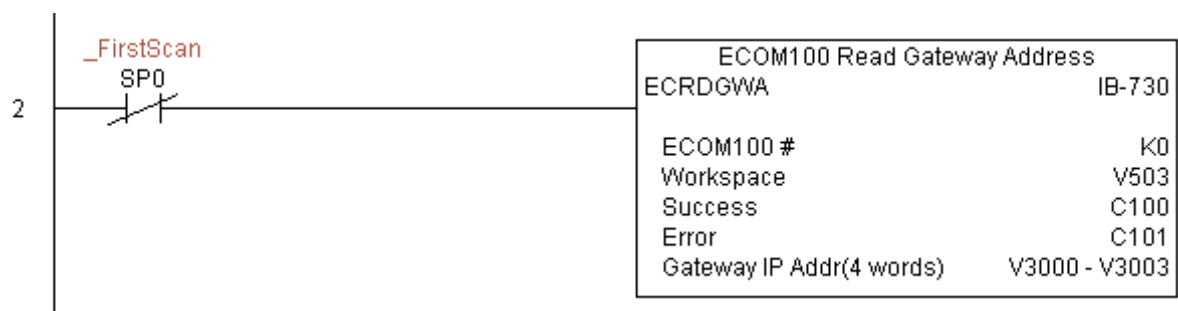
Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: На втором скан-цикле читается адрес шлюза из модуля ECOM100 и сохраняется в памяти по адресам с V3000 по V3003 (4 десятичных числа). Эти значения могут быть выведены оператору на экран монитора.

Команда ECRDGWA выполняется по переднему фронту входного условия, в этот момент выполняется чтение адреса шлюза.

При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101.



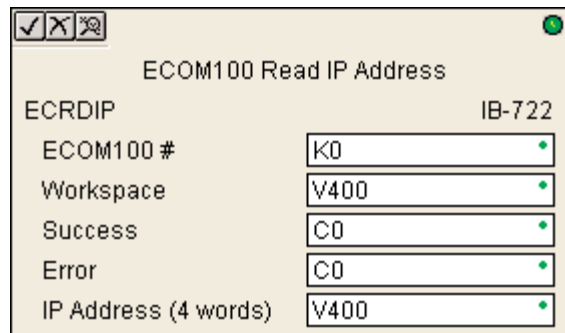
ECOM100 Read IP Address (ECRDIP) (IB-722)

Эта инструкция читает текущий IP адрес модуля ECOM100 и сохраняет его в четырех последовательных ячейках памяти в десятичном формате. Инструкция работает по переднему фронту.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).



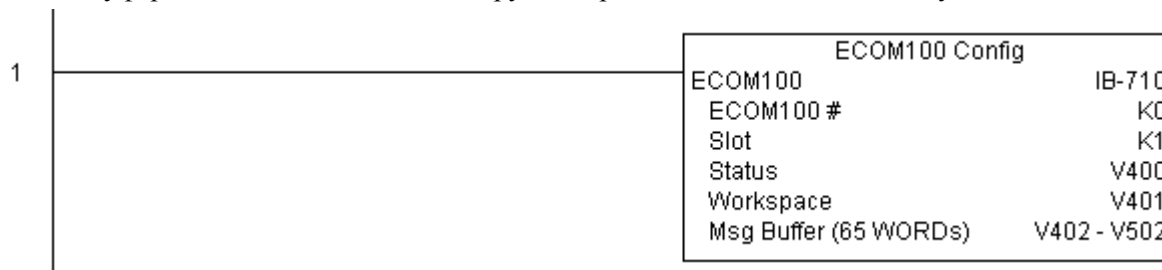
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ESxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **IP Address**: адрес первой ячейки памяти для сохранения IP адреса модуля в четырех последовательных ячейках памяти в десятичном формате.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
IP Address (4 Words) V	См. карту памяти DL06 - Слова данных

Пример:

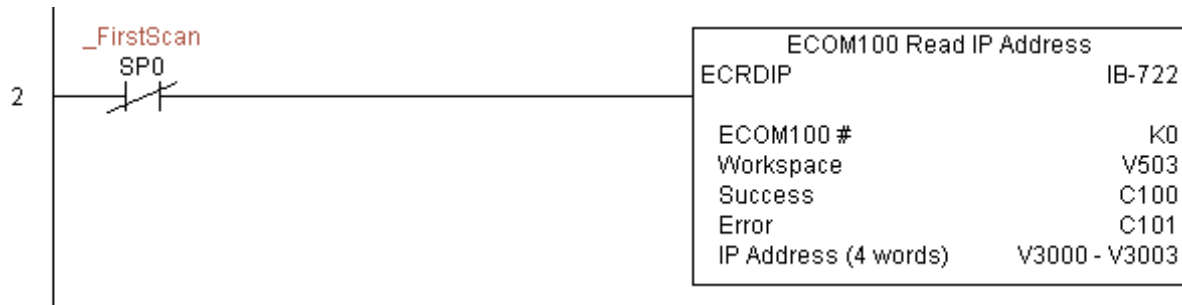
Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: На втором скан-цикле читается IP адрес модуля ECOM100 и сохраняется в памяти по адресам с V3000 по V3003 (4 десятичных числа). Эти значения могут быть выведены оператору на экран монитора.

Команда ECRDIP выполняется по переднему фронту входного условия, в этот момент выполняется чтение IP адреса.

При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101.



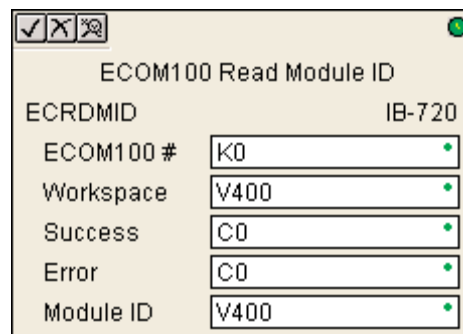
ECOM100 Read Module ID (ECRDMID) (IB-720)

Эта инструкция читает идентификатор (ID) модуля ECOM100 в формате десятичного слова (binary). Инструкция работает по переднему фронту.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).



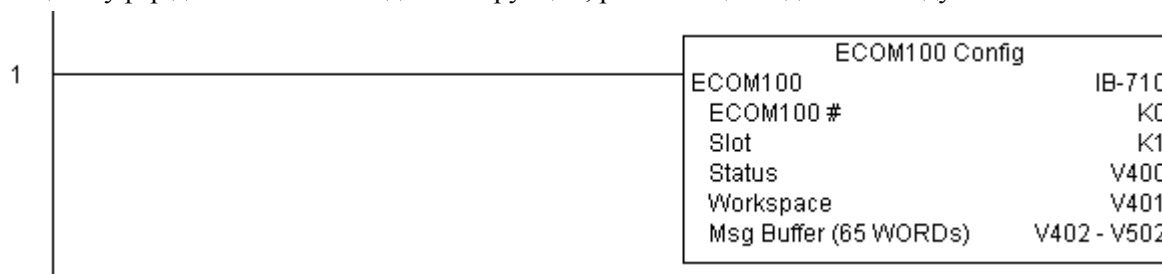
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Module ID**: адрес ячейки памяти для сохранения идентификатора (ID) модуля в виде десятичного слова (binary).

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Module ID. V	См. карту памяти DL06 - Слова данных

Пример:

Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: На втором скан-цикле читается идентификатор (ID) модуля ECOM100 и сохраняется в памяти по адресу V2000 (десятичное число).

Команда ECRDMID выполняет чтение модуля по переднему фронту входного условия. При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101.



ECOM100 Read Module Name (ECRDNAM) (IB-724)

Эта инструкция запрашивает из модуля ECOM100 его наименование, при этом задается длина запрашиваемой строки в символах. Инструкция работает по переднему фронту.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).

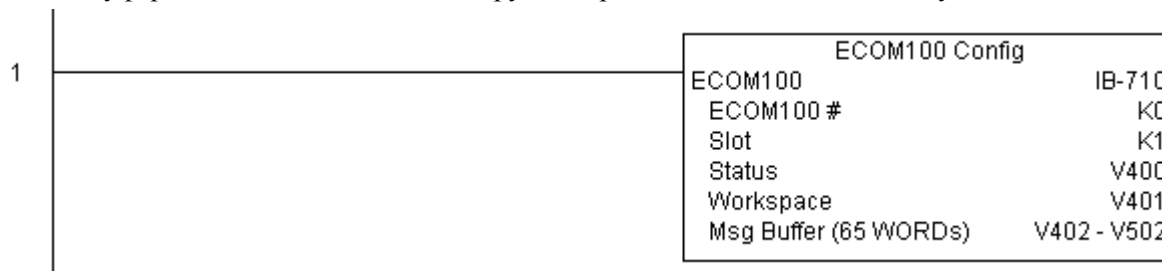
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Module Name**: адрес начальной ячейки блока памяти, в который будет сохранено наименование модуля ECOM100.
- **Num Chars**: длина строки в символах (байтах) для чтения имени модуля ECOM100.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Module Name V	См. карту памяти DL06 - Слова данных
Num Chars K	Константа: K1-128

Пример:

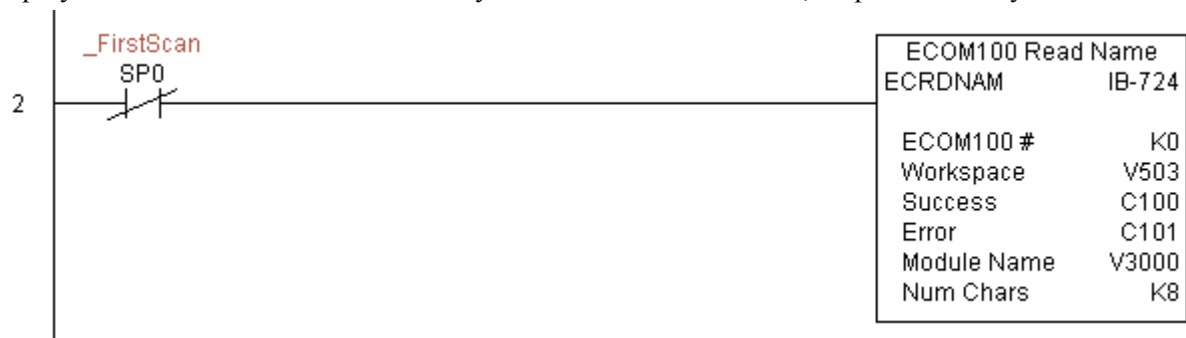
Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в каркасе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: На втором скан-цикле читается имя модуля ECOM100 и сохраняется в блоке памяти по адресам с V3000 по V3003 (8 символов). Этот текст может быть выведен на экран монитора.

Команда ECRDNAM выполняет чтение модуля по переднему фронту входного условия.

При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101.



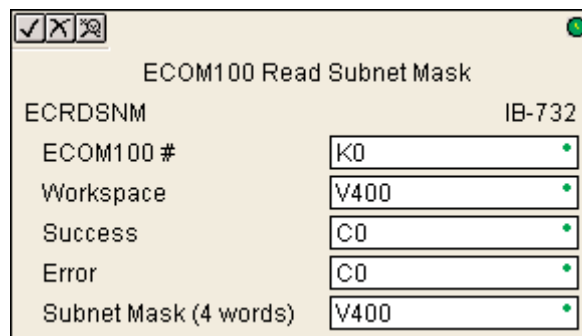
ECOM100 Read Subnet Mask (ECRDSNM) (IB-732)

Эта инструкция читает текущую маску подсети модуля ECOM100 и сохраняет ее в четырех последовательных ячейках памяти в десятичном формате. Инструкция работает по переднему фронту.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).



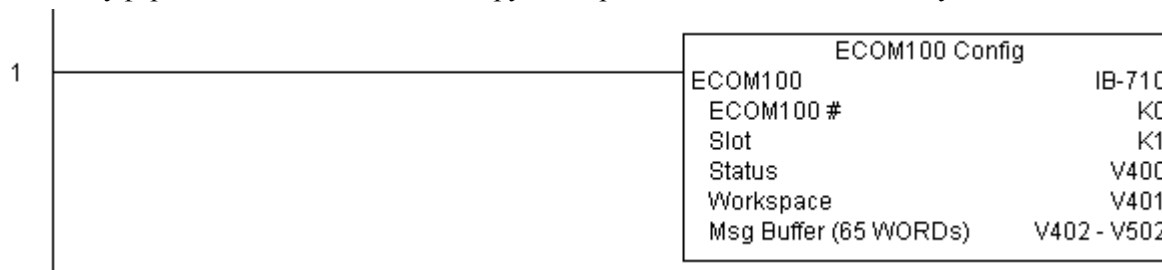
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Subnet Mask**: адрес начальной ячейки блока памяти, в который будет сохранена маска подсети в четырех последовательных ячейках памяти в десятичном формате.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Subnet Mask (4 Words) V	См. карту памяти DL06 - Слова данных

Пример:

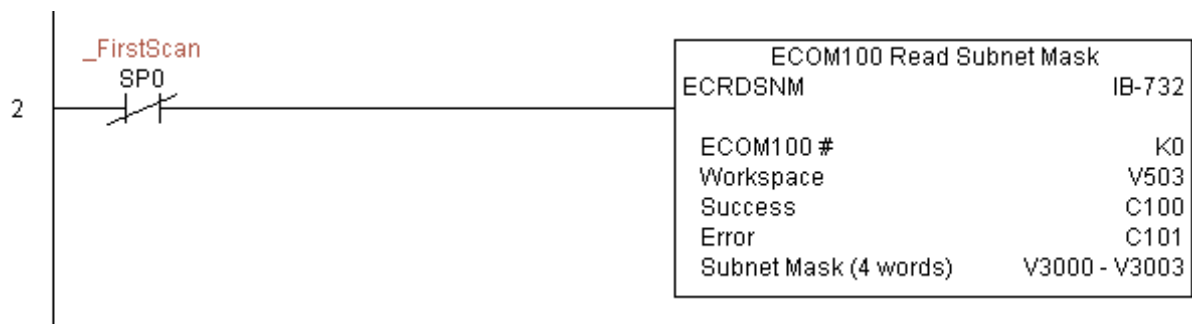
Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: На втором скан-цикле читается текущая маска подсети модуля ECOM100 и сохраняется в блоке памяти по адресам с V3000 по V3003 (4 десятичных числа). Этот текст может быть выведен на экран монитора.

Команда ECRDSNM срабатывает по переднему фронту, и в этот момент выполняет чтение маски подсети из модуля ECOM100.

При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101.



ECOM100 Write Description (ECWRDES) (IB-727)

С помощью этой команды в модуль ECOM100 записывается его описание в соответствующее поле. Команда выполняется по переднему фронту. Если требуется вставить в текст знак доллара (\$) или двойные кавычки (“”), используйте специальные последовательности, как в функциях PRINT/VPRINT, с дополнительным предстоящим знаком доллара: (\$\$) и (\$”) соответственно.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно. В случае ошибки параметр Error Code получит код ошибки модуля (меньше 100) или программы контроллера (больше 1000).

Описание модуля сохраняются во Flash ROM памяти модуля, и при выполнении этой команды модуль в момент записи остается недоступным не менее половины секунды. Тем не менее, крайне рекомендуется выполнять эту команду только один раз в первом скан-цикле программы. Эта команда срабатывает по переднему фронту, поэтому используйте нормально закрытое реле SP0 (STR NOT First Scan) вместе с командой в программе.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).

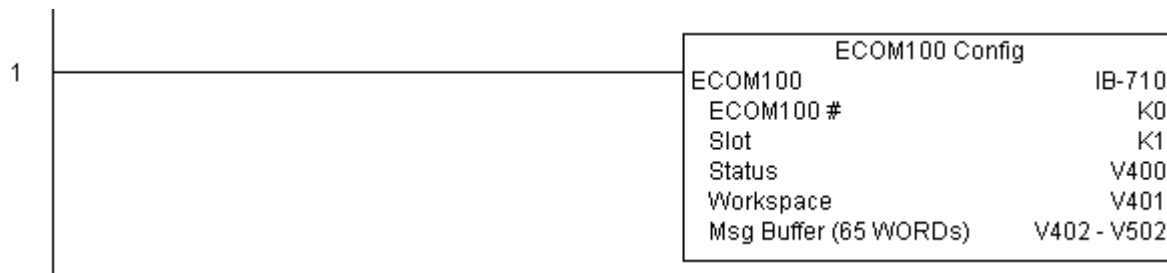
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при успешном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Error Code**: адрес ячейки памяти для сохранения кода ошибки при записи.
- **Description**: описание модуля.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Error Code V	См. карту памяти DL06 - Слова данных
Description V	Текст

Пример:

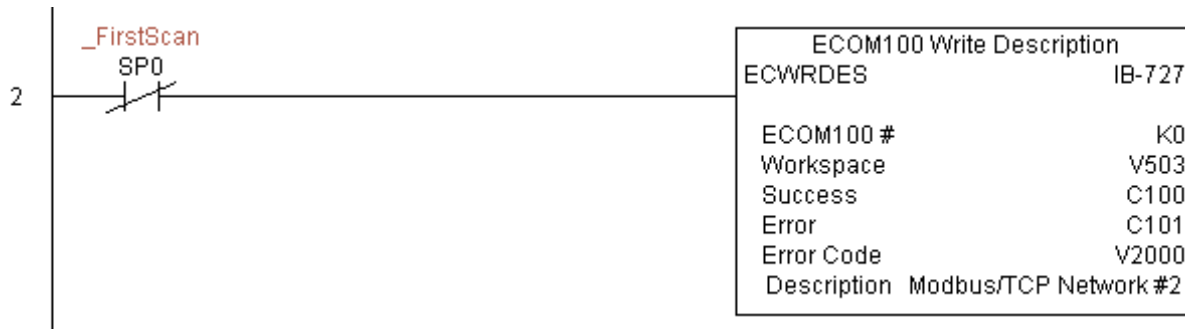
Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: На втором скан-цикле записывается строка описания модуля во Flash ROM память ECOM100. Обычно это делают в программе NetEdit, но из программы контроллера это также можно сделать с помощью этой команды.

Команда EWRDES выполняется по переднему фронту входного условия, в этот момент выполняется запись строки описания.

При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101. При ошибке ее код может быть получен в ячейке V2000.



ECOM100 Write Gateway Address (ECWRGWA) (IB-731)

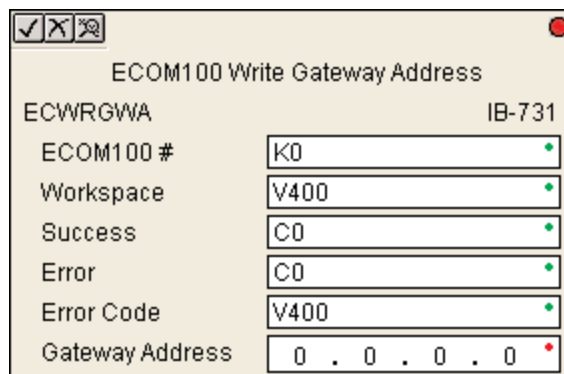
С помощью этой команды в модуль ECOM100 записывается заданный IP адрес шлюза. Команда работает по переднему фронту входного условия. См. также команду ECOM100 IP Setup (ECIPSUP) для настройки всех TCP/IP параметров: IP адрес, маска подсети, адрес шлюза.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно. В случае ошибки параметр Error Code получит код ошибки модуля (меньше 100) или программы контроллера (больше 1000).

IP адрес шлюза сохраняется во Flash ROM памяти модуля, и при выполнении этой команды модуль в момент записи остается недоступным не менее половины секунды. Тем не менее, крайне рекомендуется выполнять эту команду только один раз в первом скан-цикле программы. Эта команда срабатывает по переднему фронту, поэтому используйте нормально закрытое реле SP0 (STR NOT First Scan) вместе с командой в программе.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).



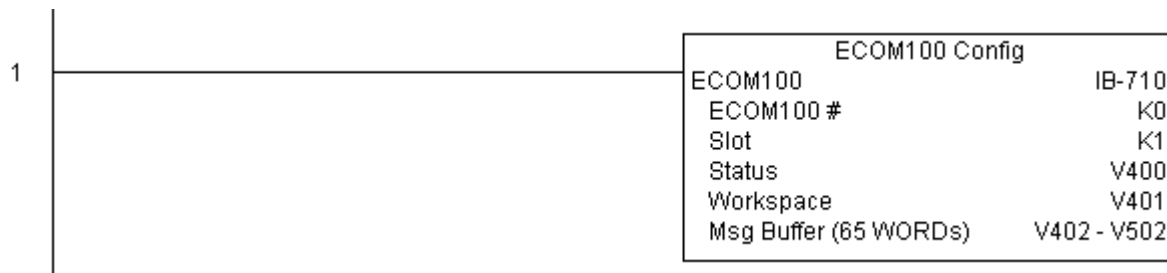
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Error Code**: адрес ячейки памяти для сохранения кода ошибки при записи.
- **Gateway Address**: IP адрес шлюза, который будет записан в память модуля ECOM100.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Error Code V	См. карту памяти DL06 - Слова данных
Gateway Address	0.0.0.1. to 255.255.255.254

Пример:

Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.

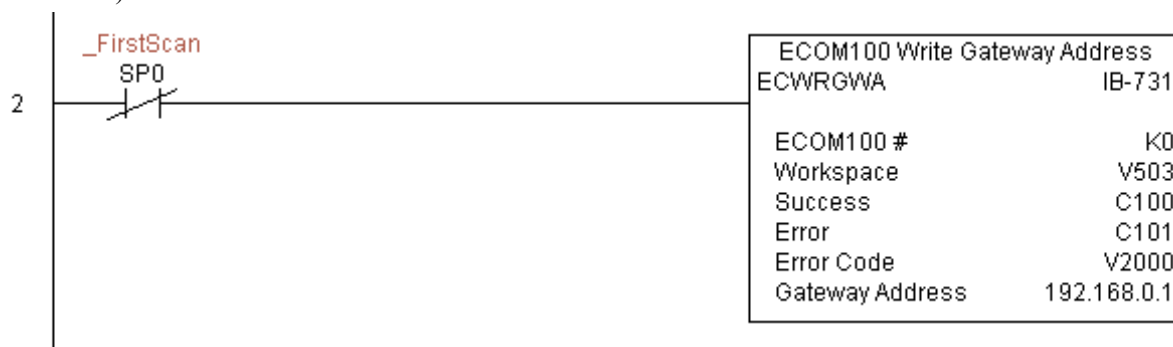


Цепь 2: На втором скан-цикле задается адрес шлюза 192.168.0.1, и он записывается во Flash ROM память модуля ECOM100.

Команда ECWRGWA выполняется по переднему фронту входного условия, в этот момент адрес шлюза посылается в модуль ECOM.

При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101. При ошибке ее код может быть получен в ячейке V2000.

Для конфигурирования всех TCP/IP параметров одной командой используйте ECOM100 IP Setup (ECIPSUP).



ECOM100 Write IP Address (ECWRIP) (IB-723)

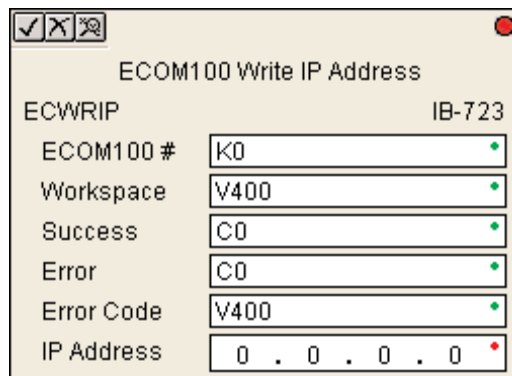
С помощью этой команды в модуль ECOM100 записывается заданный статический IP адрес. Команда работает по переднему фронту входного условия. См. также команду ECOM100 IP Setup (ECIPSUP) для настройки всех TCP/IP параметров: IP адрес, маска подсети, адрес шлюза.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно. В случае ошибки параметр Error Code получит код ошибки модуля (меньше 100) или программы контроллера (больше 1000).

IP адрес модуля сохраняется во Flash ROM памяти модуля, и при выполнении этой команды модуль в момент записи остается недоступным не менее половины секунды. Тем не менее, крайне рекомендуется выполнять эту команду только один раз в первом скан-цикле программы. Эта команда срабатывает по переднему фронту, поэтому используйте нормально закрытое реле SP0 (STR NOT First Scan) вместе с командой в программе.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).



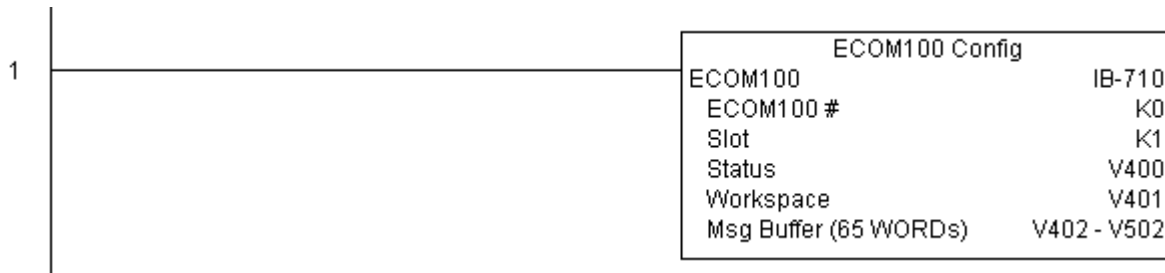
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Error Code**: адрес ячейки памяти для сохранения кода ошибки при записи.
- **IP Address**: статический IP адрес модуля, который будет записан в память модуля ECOM100.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Error Code V	См. карту памяти DL06 - Слова данных
IP Address	0.0.0.1. to 255.255.255.254

Пример:

Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: На втором скан-цикле задается статический IP адрес модуля 192.168.12.100, и записывается во Flash ROM память модуля ECOM100.

Команда ECWRIP выполняется по переднему фронту входного условия, в этот момент IP адрес записывается в модуль ECOM.

При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101. При ошибке ее код может быть получен в ячейке V2000.

Для конфигурирования всех TCP/IP параметров одной командой используйте ECOM100 IP Setup (ECIPSUP).



ECOM100 Write Module ID (ECWRMID) (IB-721)

Команда записывает в модуль ECOM100 его идентификатор (ID). Команда работает по переднему фронту входного условия. Если идентификатор модуля задан микропереключателями на плате, то эта команда не выполнится и выдаст ошибку 1005 (десятич.).

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно. В случае ошибки параметр Error Code получит код ошибки модуля (меньше 100) или программы контроллера (больше 1000).

Идентификатор модуля сохраняется во Flash ROM памяти модуля, и при выполнении этой команды модуль в момент записи остается недоступным не менее половины секунды. Тем не менее, крайне рекомендуется выполнять эту команду только один раз в первом скан-цикле программы. Эта команда срабатывает по переднему фронту, поэтому используйте нормально закрытое реле SP0 (STR NOT First Scan) вместе с командой в программе.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).

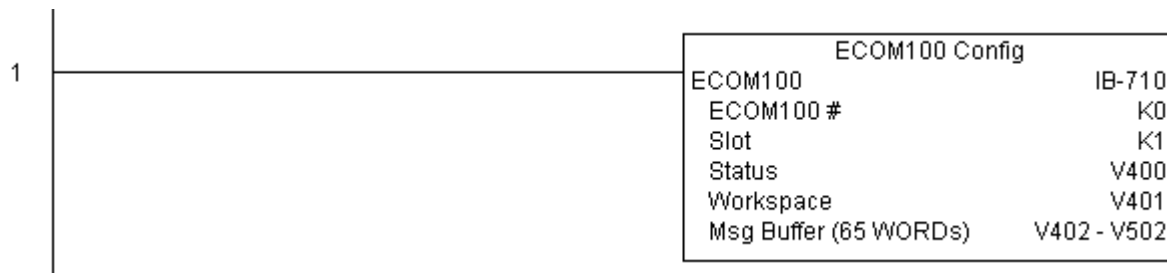
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ESxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Error Code**: адрес ячейки памяти для сохранения кода ошибки при записи.
- **Module ID**: идентификатор модуля, который должен быть записан.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Error Code V	См. карту памяти DL06 - Слова данных
Module ID	K0-65535

Пример:

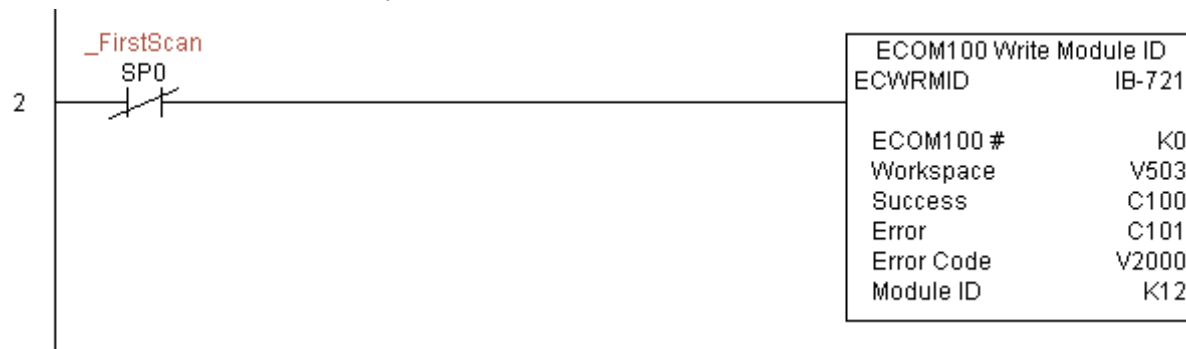
Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в каркасе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: На втором скан-цикле записывается идентификатор (ID) модуля ECOM100 во Flash ROM память. Обычно это делается программой NetEdit, но может быть также сделан и с помощью этой команды из программы контроллера.

Команда EWRMID выполняется по переднему фронту входного условия, в этот момент выполняется запись идентификатора в модуль ECOM100.

При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101. При ошибке ее код может быть получен в ячейке V2000.



ECOM100 Write Name (ECWRNAM) (IB-725)

Команда записывает в модуль ECOM100 его наименование. Команда работает по переднему фронту входного условия. Если требуется вставить в текст знак доллара (\$) или двойные кавычки (“), используйте специальные последовательности, как в функциях PRINT/VPRINT, с дополнительным предстоящим знаком доллара: (\$\$) и (\$”) соответственно.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно. В случае ошибки параметр Error Code получит код ошибки модуля (меньше 100) или программы контроллера (больше 1000).

Наименование модуля сохраняется во Flash ROM памяти модуля, и при выполнении этой команды модуль в момент записи остается недоступным не менее половины секунды. Тем не менее, крайне рекомендуется выполнять эту команду только один раз в первом скан-цикле программы. Эта команда срабатывает по переднему фронту, поэтому используйте нормально закрытое реле SP0 (STR NOT First Scan) вместе с командой в программе.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).

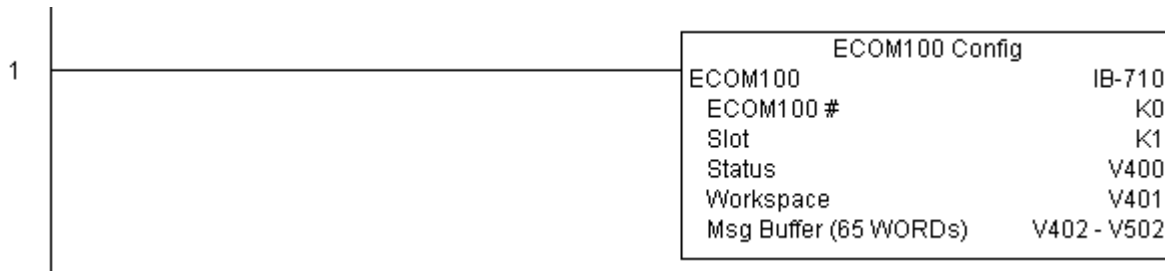
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Error Code**: адрес ячейки памяти для сохранения кода ошибки при записи.
- **Module Name**: наименование модуля, которое должно быть записано в его память.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Error Code V	См. карту памяти DL06 - Слова данных
Module Name	Текст

Пример:

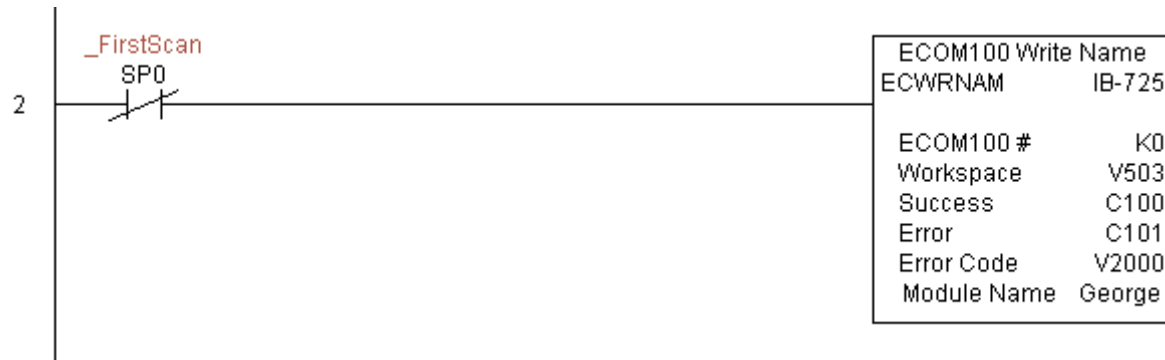
Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: На втором скан-цикле записывается наименование модуля ECOM100 во Flash ROM память. Обычно это делается программой NetEdit, но может быть также сделан и с помощью этой команды из программы контроллера.

Команда EWRNAM выполняется по переднему фронту входного условия, в этот момент наименование модуля записывается в модуль ECOM100.

При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101. При ошибке ее код может быть получен в ячейке V2000.



ECOM100 Write Subnet Mask (ECWRSNM) (IB-733)

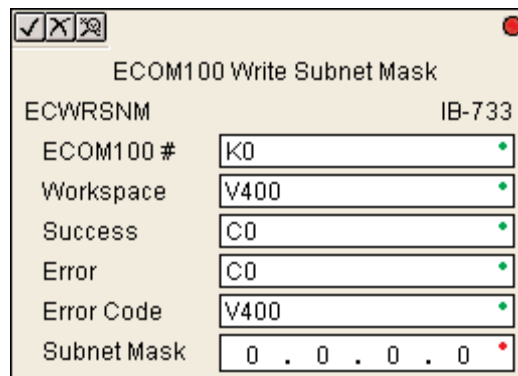
Команда записывает в модуль ECOM100 новую маску подсети. Команда работает по переднему фронту входного условия. См. также команду ECOM100 IP Setup (ECIPSUP) для настройки всех TCP/IP параметров: IP адрес, маска подсети, адрес шлюза.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При удачном или неудачном завершении команды устанавливается бит Success или Error соответственно. В случае ошибки параметр Error Code получит код ошибки модуля (меньше 100) или программы контроллера (больше 1000).

Маска подсети модуля сохраняется во Flash ROM памяти модуля, и при выполнении этой команды модуль в момент записи остается недоступным не менее половины секунды. Тем не менее, крайне рекомендуется выполнять эту команду только один раз в первом скан-цикле программы. Эта команда срабатывает по переднему фронту, поэтому используйте нормально закрытое реле SP0 (STR NOT First Scan) вместе с командой в программе.

Для работы этой команды необходимо чтобы микропереключатель 7 на интерфейсном модуле был включен (On).



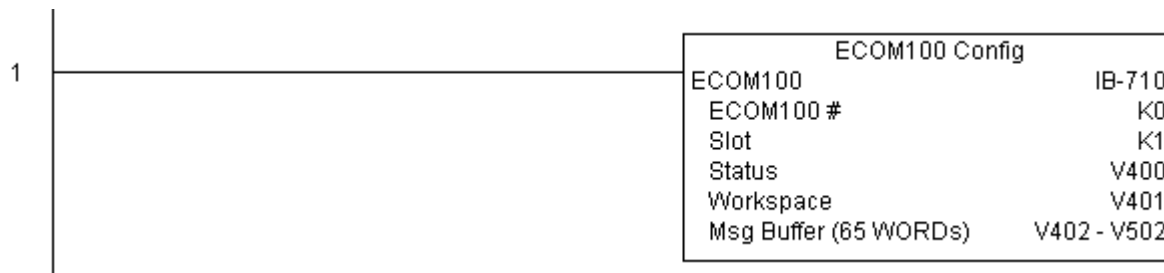
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Success**: адрес бита, который устанавливается (On) при удачном выполнении команды.
- **Error**: адрес бита, который устанавливается (On) при неудачном выполнении команды.
- **Error Code**: адрес ячейки памяти для сохранения кода ошибки при записи.
- **Subnet Mask**: маска подсети модуля, которая должна быть записана в его память.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06
Error Code V	См. карту памяти DL06 - Слова данных
Subnet Mask	Маска подсети

Пример:

Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в корпусе, то потребуется изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.



Цепь 2: На втором скан-цикле записывается маска подсети 255.255.0.0 модуля ECOM100 во Flash ROM память. Обычно это делается программой NetEdit, но может быть также сделан и с помощью этой команды из программы контроллера.

Команда ECWRSNM выполняется по переднему фронту входного условия, в этот момент маска подсети записывается в модуль ECOM100.

При успешном выполнении команды устанавливается бит C100, в противном случае – бит C101. При ошибке ее код может быть получен в ячейке V2000.

Для конфигурирования всех TCP/IP параметров одной командой используйте ECOM100 IP Setup (ECIPSUP).



ECOM100 RX Network Read (ECRX) (IB-740)

Для упрощения сетевых коммуникаций эта команда выполняется инструкцией RX (чтения) вместе с другими инструкциями чтения ECOM100 RX (ECRX) и записи ECOM100 WX (ECWX) в вашей программе. Инструкция RX (чтения) выполняется для модуля ECOM100, номер которого задан параметром ECOM100#, соответствующей командой ECOM100 Configuration в программе ранее.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При работе этой команды выполняется чтение элемента данных из указанного ведомого источника (Src) в заданный блок памяти (Dest), при этом могут быть выполнены и другие команды ECOM100 RX и ECOM100 WX, связанные с тем же модулем ECOM100#.

Например, если вы хотите прочитать данные непрерывно из пяти различных ведомых источников, вы можете поместить все 5 этих инструкций в одну цепь программы под одно условие SP1 (всегда On). Тогда они все выполнятся одна за другой автоматически.

выполняет инструкцию RX (чтения) вместе с

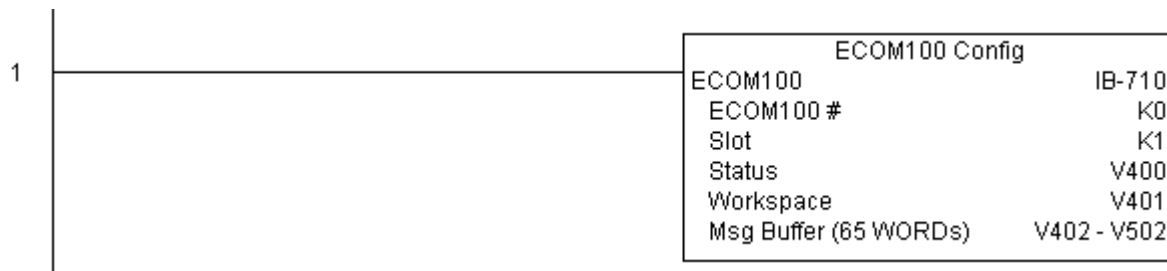
Параметры:

- **ECOM100#**: логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Slave ID**: номер ведомого модуля ECOM(100) для инструкции ECRX.
- **From Slave Element (Src)**: адрес ведомого источника данных для чтения.
- **Number of Bytes**: число читаемых байт.
- **To Master Element (Dest)**: адрес ячейки памяти ведущего приемника для сохранения читаемых данных.
- **Success**: адрес бита, который будет установлен при успешном выполнении команды.
- **Error**: адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Slave ID K	K0-90
From Slave Element (Src) X,Y,C,S,T,CT,GX,GY,V	См. карту памяти DL06
Number of Bytes K	K1-128
To Master Element (Dest) V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в корпусе, то потребуются изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.

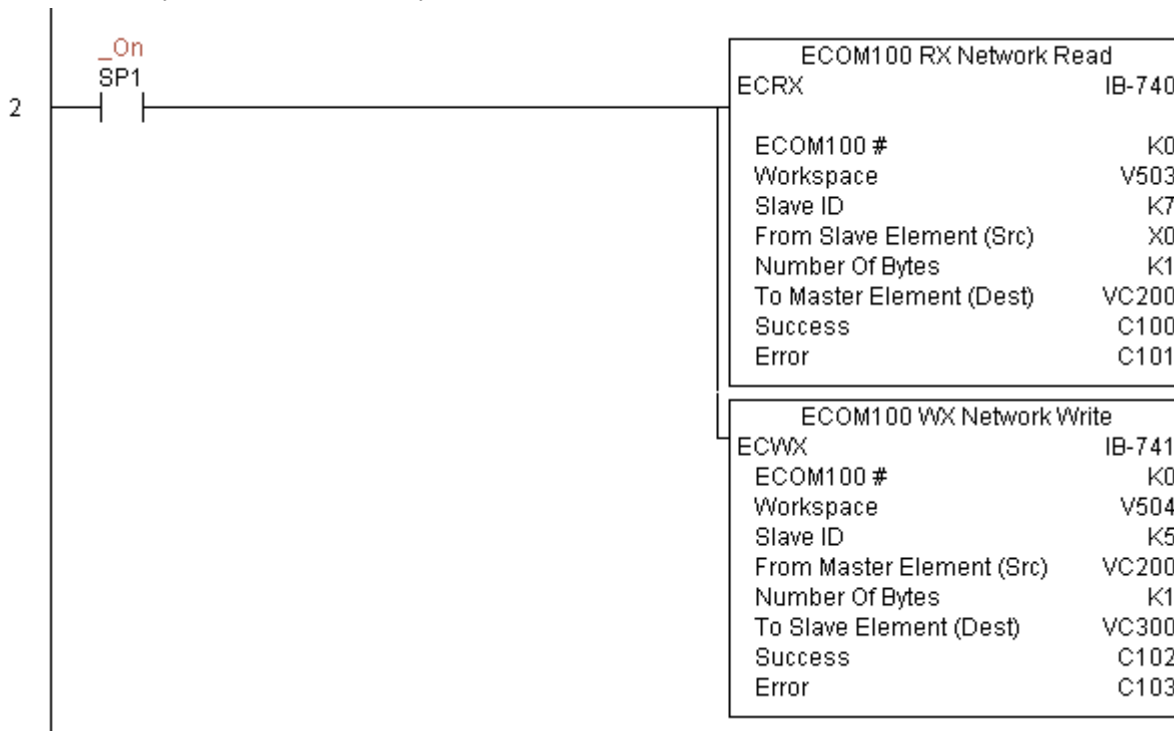


Цепь 2: Используя модуль с номером K0, выполняется чтение ячеек X0-X7 из ведомого источника с номером K7, и максимально быстро записывается в ведомый приемник с номером K5. Прочитанные данные сохраняются в локальном контроллере в C200-C207 и записываются в приемнике в ячейки C300-C307.

Обе команды ECRX и ECWX работают совместно с ECOM100 Config чтобы упростить сетевые обращения через один ресурс. Они также предоставляют упрощенные отчеты об ошибках. Теперь нет необходимости беспокоиться о проверке состоянии занятости и статуса ошибки ("busy bits", "error bits"), или помнить о номере слота модуля, и заводить счетчики или регистры сдвига для организации взаимодействия с модулем.

В этом примере SP1 (всегда On) управляет двумя командами ECRX и ECWX в одной цепи программы. В одном скан-цикле как только заканчивает выполнение команда чтения, начинает выполняться команда записи. Как только команда записи завершится, все незаконченные операции ниже в программе будут выполнены. Если нет незаконченных операций под ECWX, которые обращаются к ECOM100, то непосредственно в следующий скан-цикл команда ECRX снова начнет свой запрос.

Использование команд ECRX и ECWX для всех операций чтения и записи через модуль ECOM100 обеспечит наиболее быстрые сетевые коммуникации. Для локальных последовательных портов, модулей DCM или модулей ECOM используйте команды NETCFG и NETRX/NETWX.



ECOM100 WX Network Write(ECWX) (IB-741)

Для упрощения сетевых коммуникаций эта команда выполняется вместе с другими инструкциями чтения ECOM100 RX (ECRX) и записи ECOM100 WX (ECWX) в вашей программе. Инструкция WX (записи) выполняется для модуля ECOM100, номер которого задан параметром ECOM100#, соответствующей командой ECOM100 Configuration в программе ранее.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При работе этой команды выполняется запись элемента данных из буфера V-памяти, указанного ведущего источника (Src), в заданный блок памяти ведомого приемника (Dest), при этом могут быть выполнены и другие команды ECOM100 RX и ECOM100 WX, связанные с тем же модулем ECOM100#.

Например, если вы хотите прочитать и записать данные непрерывно в пять различных ведомых приемников, вы можете поместить все 5 этих инструкций в одну цепь программы под одно условие SP1 (всегда On). Тогда они все выполняются одна за другой автоматически.

выполняет инструкцию WX (записи) вместе с

Параметры:

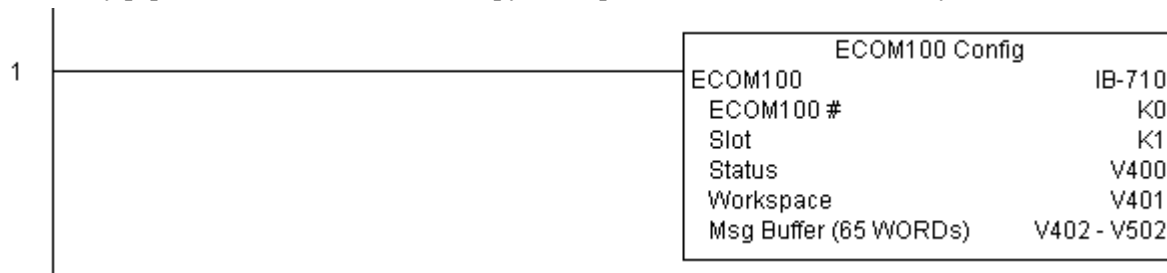
- **ECOM100#:** логический номер модуля, присваиваемый конкретному модулю ECOM100. Другие команды ECxxxx, работающие с этим модулем должны адресоваться на этот номер.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Slave ID:** номер ведомого модуля ECOM(100) для инструкции ECWX.
- **From Master Element (Src):** адрес ячейки памяти ведущего источника ECOM100, откуда будут браться данные.
- **Number of Bytes:** число байт, записываемых в ведомый приемник.
- **To Slave Element (Dest):** адрес ведомого приемника данных.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
ECOM100# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Slave ID K	K0-90
From Master Element (Src) V	См. карту памяти DL06 - Слова данных
Number of Bytes K	K1-128
To Slave Element (Dest) . . X,Y,C,S,T,CT,GX,GY,V	См. карту памяти DL06
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

Цепь 1: Команда ECOM100 Config отвечает за координацию/блокировку всех инструкций, обращающихся к одному модулю ECOM100. Логический номер модуля, установленного в слот 1, задан как K0. Все инструкции ECOM100 ссылаются на этот номер как K0. Если необходимо переместить модуль в другой слот в корпусе, то потребуются изменить только этот номер в одной инструкции ECOM100 Config. Ячейка V400 используется как общий для всех инструкций регистр статуса модуля. Ячейка V401 используется

для координации логики работы всех команд, работающих с этим модулем. Область памяти V402-V502 – общий буфер длиной 130 байт для инструкций, работающих с данным модулем ECOM100.

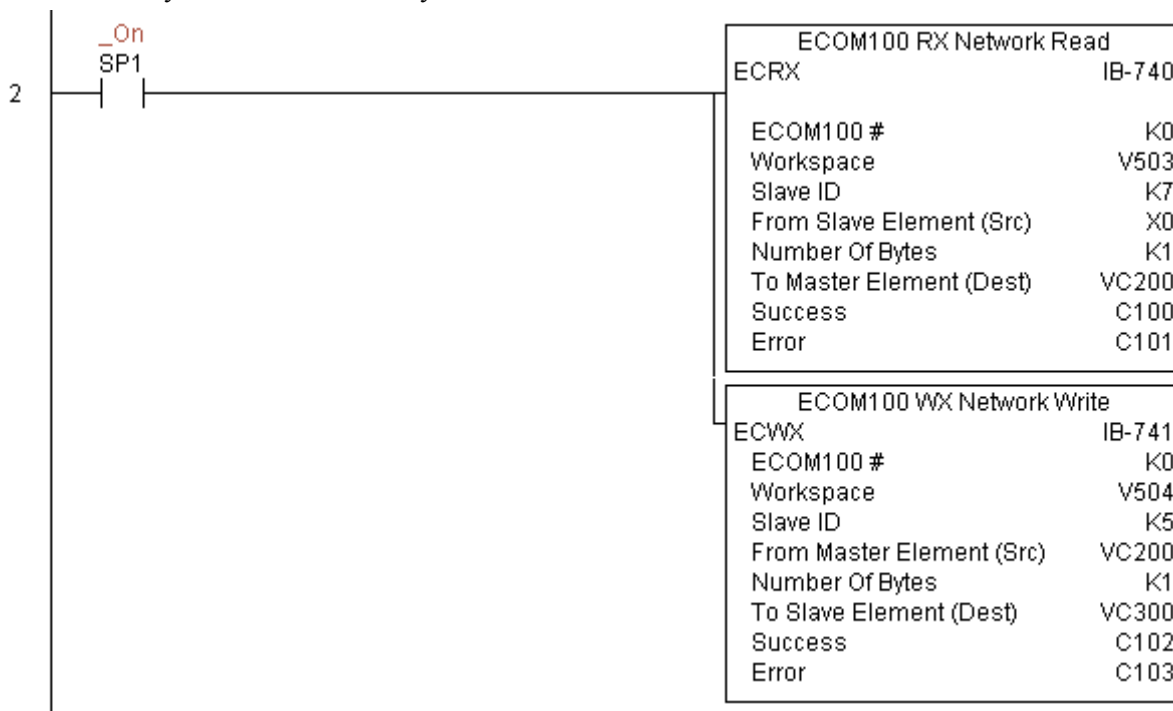


Цепь 2: Используя модуль с номером K0, выполняется чтение ячеек X0-X7 из ведомого источника с номером K7, и максимально быстро записывается в ведомый приемник с номером K5. Прочитанные данные сохраняются в локальном контроллере в C200-C207 и записываются в приемнике в ячейки C300-C307.

Обе команды ECRX и ECWX работают совместно с ECOM100 Config чтобы упростить сетевые обращения через один ресурс. Они также предоставляют упрощенные отчеты об ошибках. Теперь нет необходимости беспокоиться о проверке состоянии занятости и статуса ошибки ("busy bits", "error bits"), или помнить о номере слота модуля, и заводить счетчики или регистры сдвига для организации взаимодействия с модулем.

В этом примере SP1 (всегда On) управляет двумя командами ECRX и ECWX в одной цепи программы. В одном скан-цикле как только заканчивается выполнение команда чтения, начинает выполняться команда записи. Как только команда записи завершится, все незаконченные операции ниже в программе будут выполнены. Если нет незаконченных операций под ECWX, которые обращаются к ECOM100, то непосредственно в следующий скан-цикл команда ECRX снова начнет свой запрос.

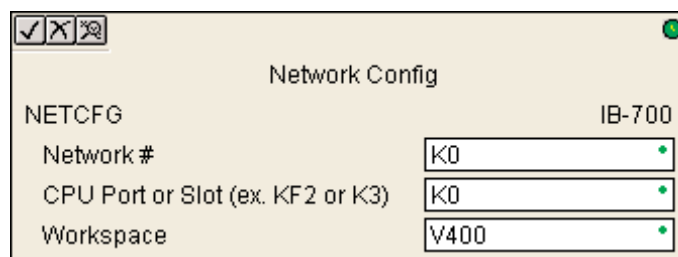
Использование команд ECRX и ECWX для всех операций чтения и записи через модуль ECOM100 обеспечит наиболее быстрые сетевые коммуникации. Для локальных последовательных портов, модулей DCM или модулей ECOM используйте команды NETCFG и NETRX/NETWX.



NETCFG Network Configuration (NETCFG) (IB-700)

Эта команда предопределяет всю необходимую информацию для выполнения чтения/записи (RX/WX) при использовании инструкций NETRX и NETWX через локальный последовательный порт, DCM или ECOM модуль.

Эта команда должна находиться в самом начале программы контроллера, вместе с другими конфигурационными инструкциями.



Если в вашем контроллере более одного локального последовательного порта DCM или ECOM, то следует использовать отдельные команды конфигурирования для каждой RX/WX сети, которые используют инструкции NETRX/NETWX.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

Второй параметр «CPU Port or Slot» является таким же как и старший байт первой LD инструкции если вы писали цепи RX или WX самостоятельно. Это значение индивидуально для CPU и порта (см. инструкцию пользователя), но возможные значения могут быть такие: KF2 – для 2 локального порта CPU 06, K3 – для DCM или ECOM модулей в 3 слоте 205 каркаса или K37 для модуля DCM в 3 дополнительном каркасе 405, в слоте 7.

Параметры:

- **Network#:** уникальный номер для каждой DCM или ECOM сети.
- **CPU Port or Slot:** номер порта CPU или номер слота, в который установлен модуль.
- **Workspace:** адрес ячейки памяти, который будет использован командой.

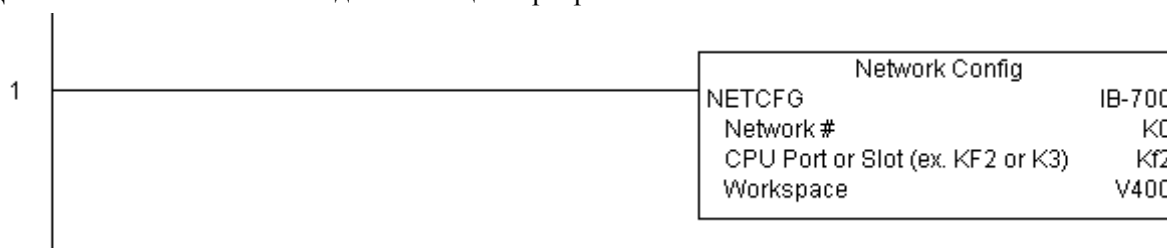
Параметр	Диапазон DL06
Network# K	Константа: K0-255
CPU Port or Slot K	Константа: K0-FF
Workspace V	См. карту памяти DL06 - Слова данных

Пример:

Инструкция Network Configuration координирует все взаимодействия с другими коммуникационными инструкциями (NETRX/NETWX). Для каждой сети на основе последовательного порта, DCM или ECOM модуля должна быть отдельная инструкция Network Configuration. Все они должны быть в самом начале программы и выполняться в каждом программном цикле.

Эта команда определяет номер сети как K0 для локального последовательного порта №2 на CPU (KF2). Для адресации локального последовательного порта на CPU или модулей DCM/ECOM используйте такое же значение, которое используется в старшем значащем байте первой LD инструкции в обычной цепи RX/WX. В каждой команде NETRX or NETWX, работающей с этой сетью, нужно также указывать K0 в параметре Network#.

Регистр Workspace используется для хранения информации о состоянии порта или модуля, распределяя аппаратный ресурс между командами NETRX и NETWX. Ячейка памяти, занимаемая этим регистром НЕ ДОЛЖНА использоваться где-либо еще в программе.



Network RX Read (NETRX) (IB-701)

Для упрощения сетевых коммуникаций эта команда выполняется вместе с другими инструкциями чтения и записи (Network RX (NETRX), Network WX (NETWX)) в вашей программе. Инструкция RX (чтения) выполняется для указанной в параметре Network# сети, которая сконфигурирована соответствующей командой Network Configuration (NETCFG) в программе ранее.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При работе этой команды выполняется чтение элемента данных из указанного ведомого источника (Src) в заданный блок памяти (Dest), при этом могут быть выполнены и другие команды Network RX и Network WX, связанные с этой же сетью Network#.

Например, если вы хотите прочитать данные непрерывно из пяти различных ведомых источников, вы можете поместить все 5 этих инструкций в одну цепь программы под одно условие SP1 (всегда On). Тогда они все выполняются одна за другой автоматически.

выполняет инструкцию RX (чтения) вместе с

Параметры:

- **Network#:** логический номер сети, организованной на базе одного последовательного порта CPU, модуля DCM или ECOM и ранее определенной командой NETCFG.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Slave ID:** номер ведомого контроллера для инструкции NETRX.
- **From Slave Element (Src):** адрес ведомого источника данных для чтения.
- **Number of Bytes:** число читаемых байт.
- **To Master Element (Dest):** адрес ячейки памяти ведущего приемника для сохранения читаемых данных.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
Network# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Slave ID K	K0-90
From Slave Element (Src) X,Y,C,S,T,CT,GX,GY,V	См. карту памяти DL06
Number of Bytes K	K1-128
To Master Element (Dest) V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

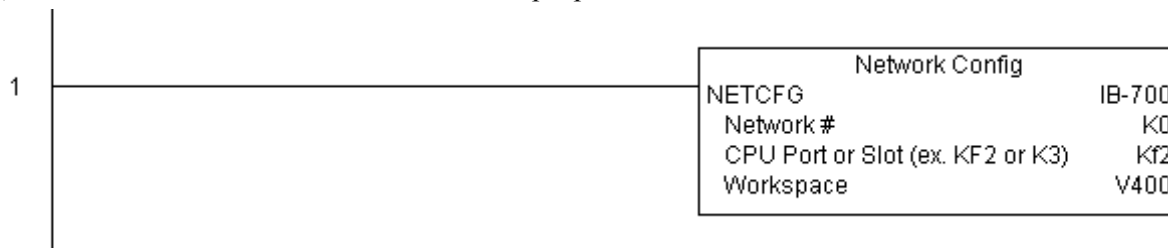
Цепь 1: Инструкция Network Configuration координирует все взаимодействия с другими коммуникационными инструкциями (NETRX/NETWX). Для каждой сети на основе последовательного порта, DCM или ECOM модуля должна быть отдельная инструкция Network Configuration. Все они должны быть в самом начале программы и выполняться в каждом программном цикле.

Эта команда определяет номер сети как K0 для локального последовательного порта №2 на CPU (KF2). Для адресации локального последовательного порта на CPU или модулей DCM/ECOM используйте такое же значение, которое используется в старшем значащем байте первой LD инструкции в обычной цепи

DirectSOFT5 Команды контроллера DL06

RX/WX. В каждой команде NETRX or NETWX, работающей с этой сетью, нужно также указывать K0 в параметре Network#.

Регистр Workspace используется для хранения информации о состоянии порта или модуля, распределяя аппаратный ресурс между командами NETRX и NETWX. Ячейка памяти, занимаемая этим регистром НЕ ДОЛЖНА использоваться где-либо еще в программе.

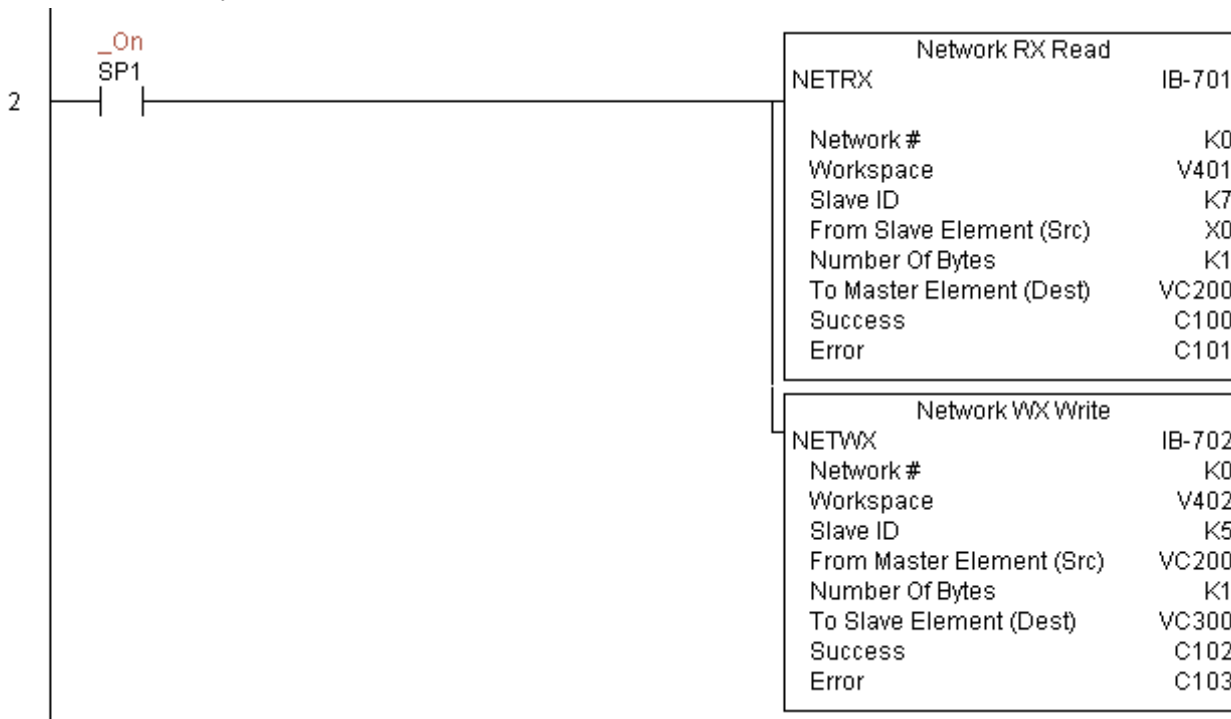


Цепь 2: Используя сеть с номером K0, выполняется чтение ячеек X0-X7 из ведомого источника с номером K7, и максимально быстро записывается в ведомый приемник с номером K5. Прочитанные данные сохраняются в локальном контроллере в C200-C207, и записываются в приемнике K5 в ячейки C300-C307.

Обе команды NETRX и NETWX работают совместно с инструкцией Network Config чтобы упростить сетевые обращения через один ресурс. Они также предоставляют упрощенные отчеты об ошибках. Теперь нет необходимости беспокоиться о проверке состоянии занятости и статуса ошибки ("busy bits", "error bits"), или помнить о номере слота модуля, и заводить счетчики или регистры сдвига для организации взаимодействия с модулем.

В этом примере SP1 (всегда On) управляет двумя командами NETRX и NETWX в одной цепи программы. В одном скан-цикле как только заканчивает выполнение команда чтения, начинается выполнение команда записи. Как только команда записи завершится, все незаконченные операции ниже в программе будут выполнены. Если нет незаконченных операций NETRX и NETWX под этой командой, то непосредственно в следующий скан-цикл команда NETRX снова начнет свой запрос.

Использование команд NETRX и NETWX для всех операций чтения и записи через последовательный порт на CPU или модули DCM/ECOM обеспечит наиболее быстрые сетевые коммуникации. Для модулей ECOM100 используйте команды ECOM100 и ECRX/ECWX.



Network WX Write (NETWX) (IB-702)

Для упрощения сетевых коммуникаций эта команда выполняется вместе с другими инструкциями чтения и записи (Network RX (NETRX), Network WX (NETWX)) в вашей программе. Инструкция WX (записи) выполняется для указанной в параметре Network# сети, которая сконфигурирована соответствующей командой Network Configuration (NETCFG) в программе ранее.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

При работе этой команды выполняется запись данных из области памяти ведущего источника в указанный ведомый приемник, при этом могут быть выполнены и другие команды Network RX и Network WX, связанные с этой же сетью Network#.

Например, если вы хотите прочитать и записать данные непрерывно в пять различных ведомых приемников, вы можете поместить все 5 этих инструкций в одну цепь программы под одно условие SP1 (всегда On). Тогда они все выполнятся одна за другой автоматически.

выполняет инструкцию WX (записи) вместе с

Параметры:

- **Network#:** логический номер сети, организованной на базе одного последовательного порта CPU, модуля DCM или ECOM и ранее определенной командой NETCFG.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Slave ID:** номер ведомого контроллера для инструкции NETWX.
- **From Master Element (Src):** адрес ячейки памяти ведущего источника для чтения данных.
- **Number of Bytes:** число записываемых байт.
- **To Slave Element (Dest):** адрес ведомого приемника данных для записи.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
Network# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Slave ID K	K0-90
From Master Element (Src) V	См. карту памяти DL06 - Слова данных
Number of Bytes K	K1-128
To Slave Element (Dest) . . X,Y,C,S,T,CT,GX,GY,V	См. карту памяти DL06
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

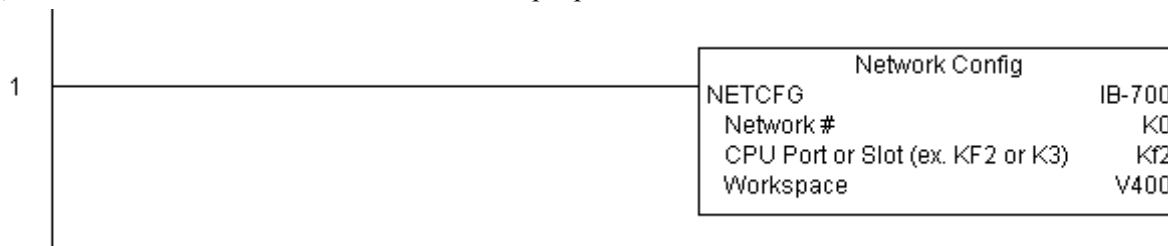
Цепь 1: Инструкция Network Configuration координирует все взаимодействия с другими коммуникационными инструкциями (NETRX/NETWX). Для каждой сети на основе последовательного порта, DCM или ECOM модуля должна быть отдельная инструкция Network Configuration. Все они должны быть в самом начале программы и выполняться в каждом программном цикле.

Эта команда определяет номер сети как K0 для локального последовательного порта №2 на CPU (KF2). Для адресации локального последовательного порта на CPU или модулей DCM/ECOM используйте такое же значение, которое используется в старшем значащем байте первой LD инструкции в обычной цепи

DirectSOFT5 Команды контроллера DL06

RX/WX. В каждой команде NETRX or NETWX, работающей с этой сетью, нужно также указывать K0 в параметре Network#.

Регистр Workspace используется для хранения информации о состоянии порта или модуля, распределяя аппаратный ресурс между командами NETRX и NETWX. Ячейка памяти, занимаемая этим регистром НЕ ДОЛЖНА использоваться где-либо еще в программе.

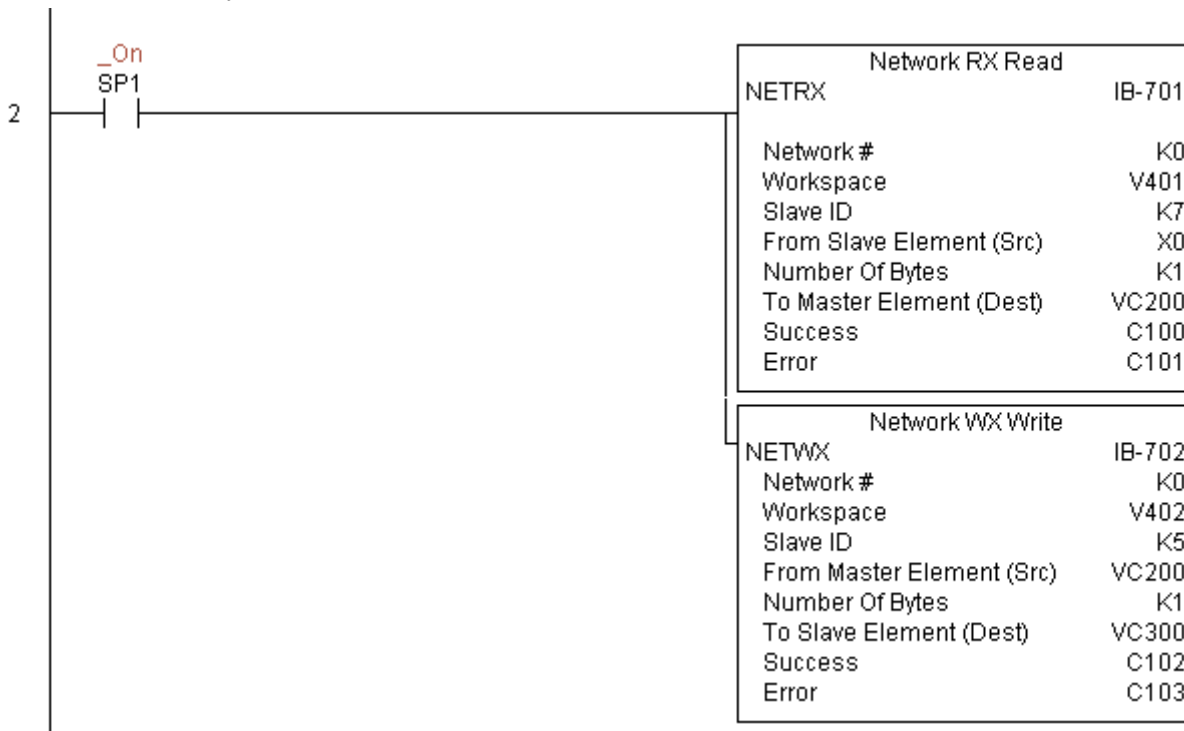


Цепь 2: Используя сеть с номером K0, выполняется чтение ячеек X0-X7 из ведомого источника с номером K7, и максимально быстро записывается в ведомый приемник с номером K5. Прочитанные данные сохраняются в локальном контроллере в C200-C207, и записываются в приемнике K5 в ячейки C300-C307.

Обе команды NETRX и NETWX работают совместно с инструкцией Network Config чтобы упростить сетевые обращения через один ресурс. Они также предоставляют упрощенные отчеты об ошибках. Теперь нет необходимости беспокоиться о проверке состоянии занятости и статуса ошибки ("busy bits", "error bits"), или помнить о номере слота модуля, и заводить счетчики или регистры сдвига для организации взаимодействия с модулем.

В этом примере SP1 (всегда On) управляет двумя командами NETRX и NETWX в одной цепи программы. В одном скан-цикле как только заканчивает выполнение команда чтения, начинается выполнение команда записи. Как только команда записи завершится, все незаконченные операции ниже в программе будут выполнены. Если нет незаконченных операций NETRX и NETWX под этой командой, то непосредственно в следующий скан-цикл команда NETRX снова начнет свой запрос.

Использование команд NETRX и NETWX для всех операций чтения и записи через последовательный порт на CPU или модули DCM/ECOM обеспечит наиболее быстрые сетевые коммуникации. Для модулей ECOM100 используйте команды ECOM100 и ECRX/ECWX.



Команды работы с модулем H0-CTRIO

CTRIO Configuration (CTRIO) (IB-1000)

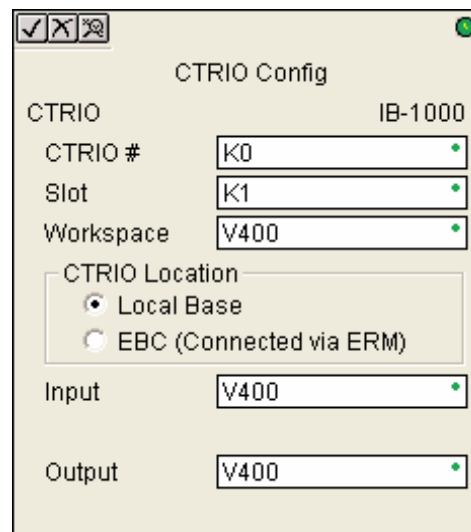
Эта команда определяет все общие параметры для одного указанного CTRIO модуля, с которым будут работать инструкции CTRIO, такие как: CTRLDPR - CTRIO Load Profile, CTREDRL - CTRIO Edit and Reload Preset Table, CTRRTLМ – CTRIO Run to Limit Mode и пр.

Параметры Input/Output могут быть скопированы непосредственно из конфигурации CTRIO Workbench этого модуля. Поскольку поведение модуля несколько отличается когда он в локальном каркасе или в EBC и подсоединен через ERM, то необходимо это указать. Контроллер DL06 поддерживает операции только с локальной базой.

Эта команда должна быть в самом начале программы вместе с другими конфигурационными инструкциями.

Если в контроллере более одного модуля CTRIO, то необходимо использовать отдельные команды CTRIO Configuration (CTRIO) для каждого модуля. И в каждой такой команде должен быть указан уникальный номер модуля CTRIO#, таким способом они дифференцируются между модулями.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.



Параметры:

- **CTRIO#**: логический уникальный номер модуля CTRIO, назначаемый пользователем.
- **Slot**: номер слота, в котором размещен модуль.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **CTRIO Location**: вариант размещения модуля, для DL06 может быть только локальный каркас.
- **Input**: здесь указывается такой же регистр, который указан в инструкции CTRIO Workbench как начальная ячейка памяти для входов (Starting V address for inputs) для этого же модуля.
- **Outputs**: здесь указывается такой же регистр, который указан в инструкции CTRIO Workbench как начальная ячейка памяти для выходов (Starting V address for outputs) для этого же модуля.

Параметр	Диапазон DL06
CTRIO# K	Константа: K0-255
Slot K	Константа: K1-4
Workspace V	См. карту памяти DL06 - Слова данных
Input V	См. карту памяти DL06 - Слова данных
Output V	См. карту памяти DL06 - Слова данных

Пример:

Цепь 1: Модуль CTRIO привязывается к слоту 2 локального каркаса. Каждый модуль CTRIO в системе должен иметь соответствующую инструкцию настройки CTRIO Configuration (CTRIO). Она должна выполняться в программе раньше других команд, обращающихся к этому модулю. Здесь модуль конфигурируется на использование памяти V2000-V2025 для помещения входных данных, и V2030-V2061 для выходных.

1

CTRIO Config	
CTRIO	IB-1000
CTRIO #	K1
Slot	K2
Workspace	V400
Input	V2000 - V2025
Output	V2030 - V2061

CTRIO Add Entry to End of Preset Table (CTRADPT) (IB-1005)

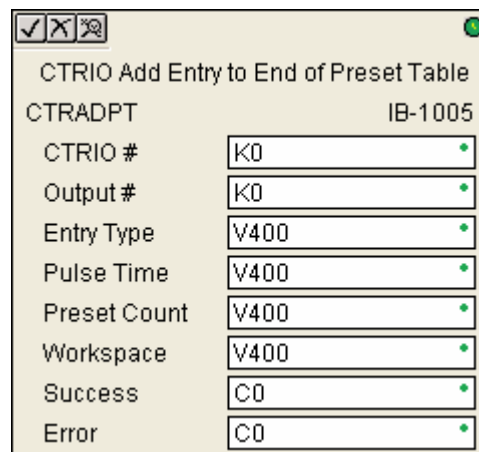
Эта команда добавляет настройку выхода модуля CTRIO в конец таблицы его настроек. Команда работает по переднему фронту. Требуется для исполнения более одного программного цикла. По результату работы команды выставляется бит Success (успешное выполнение) или Error (ошибка при выполнении). Если возникла ошибка, используйте команду CTRIO Read Error Code (CTRRDER) для получения расширенной информации о ней.

Типы настроек (установок):

- K0: Set
- K1: Reset
- K2: Pulse On (uses Pulse Time)
- K3: Pulse Off (uses Pulse Time)
- K4: Toggle
- K5: Reset Count

Заметьте. Параметр Pulse Time игнорируется некоторыми типами установок.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.



Параметры:

- **CTRIO#:** логический уникальный номер модуля CTRIO, назначаемый пользователем.
- **Output#:** номер выхода модуля CTRIO.
- **Entry Type:** тип установки.
- **Pulse Time:** время импульса для установок типа Pulse On and Pulse Off
- **Preset Count:** начальное значение счетчика, которое будет установлено после сброса.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
CTRIO# K	Константа: K0-255
Output# K	Константа: K0-3
Entry Type V,K	K0-5 и См. карту памяти DL06 - Слова данных
Pulse Time V,K	K0-65535 и См. карту памяти DL06 - Слова данных
Preset Count V,K	K0-2147434528 и См. карту памяти DL06 - Слова данных
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

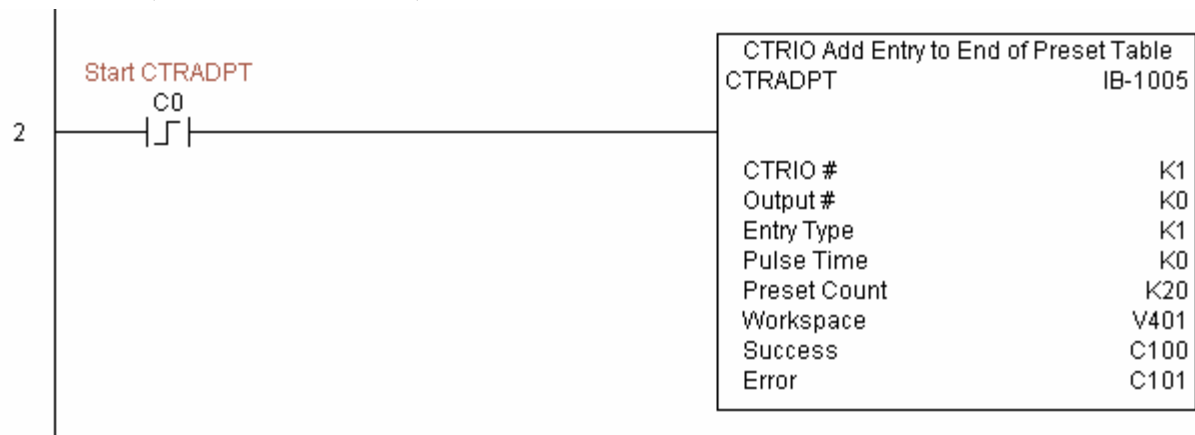
Пример:

Цепь 1: Здесь привязывается модуль CTRIO к слоту 2 локального каркаса. Для каждого модуля CTRIO нужно использовать свою отдельную команду настройки CTRIO Config перед тем как использовать другие команды CTRxxxx, работающие с этим модулем. Модуль CTRIO конфигурируется на использование памяти V2000-V2025 для входных данных, и V2030-V2061 – для выходных данных.



Цепь 2: Это пример простого использования команды CTRADPT. Условие C здесь добавлено чтобы дать возможность программисту контролировать исполнение команды в Data View. Срабатывание условия выполнит команду CTRADPT, которая добавит новую установку (preset) в таблицу настроек для выхода 0 модуля CTRIO в слоте 2. Новая установка будет управлять СБРОСОМ (тип установки K1=reset), для этого типа установки время импульса оставлено в нуле, счетчик, при котором будет сброс равен 20.

Порядок использования следующий: загрузить файл CTRADPT_ex1.cwb на модуль CTRIO, затем ввести показанную здесь программу, перейти в режим RUN, в Data View включить выход 0 переводя бит C2 в значение On, переключить кодировщик на CTRIO на значение выше 10, и тогда индикатор выхода 0 загорится и будет гореть для всех счетов больше 10. Потом сбросьте счетчик с помощью C1, включите C0 чтобы выполнить команду CTRADPT, которая добавит сброс для выхода 0 на счет 20, включите выход 0 посредством C2, затем включите кодировщик на 10+ (выход 0 должен включиться) и затем продолжайте счет до 20+ (выход 0 выключится).



Цепь 3: Эта цепь позволяет программисту сбросить счетчик из программы.



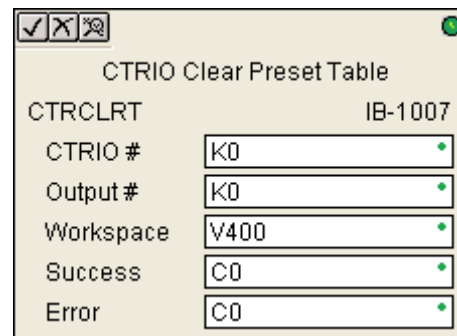
Цепь 4: Эта цепь позволяет оператору включить выход 0 из программы.



CTRIO Clear Preset Table (CTRCLRT) (IB-1007)

Эта команда очищает таблицу настроек модуля CTRIO по переднему фронту. Требуется для выполнения более одного программного цикла. По результату работы команды выставляется бит Success (успешное выполнение) или Error (ошибка при выполнении). Если возникла ошибка, используйте команду CTRIO Read Error Code (CTRRDER) для получения расширенной информации о ней.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.



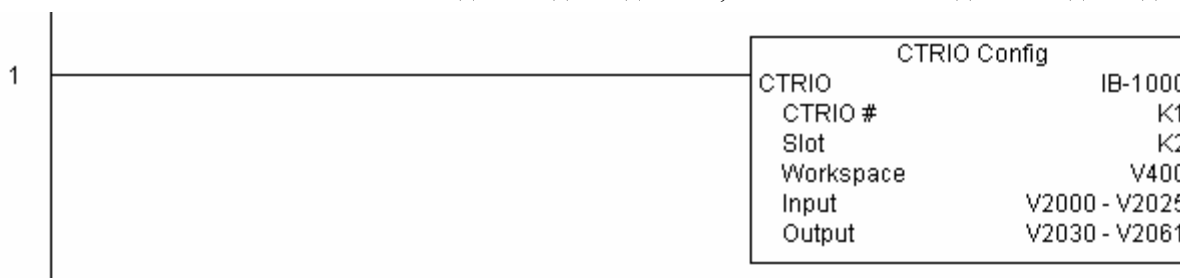
Параметры:

- **CTRIO#:** логический уникальный номер модуля CTRIO, назначаемый пользователем.
- **Output#:** номер выхода модуля CTRIO.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
CTRIO# K	Константа: K0-255
Output# K	Константа: K0-3
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

Цепь 1: Здесь привязывается модуль CTRIO к слоту 2 локального каркаса. Для каждого модуля CTRIO нужно использовать свою отдельную команду настройки CTRIO Config перед тем как использовать другие команды CTRxxxx, работающие с этим модулем. Модуль CTRIO конфигурируется на использование памяти V2000-V2025 для входных данных, и V2030-V2061 – для выходных данных.

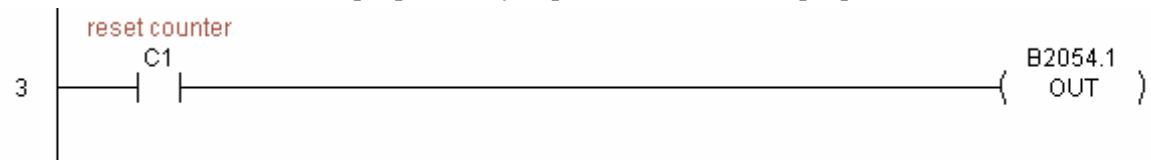


Цепь 2: Это пример простого использования команды CTRCLRT. Условие C0 здесь добавлено чтобы дать возможность программисту контролировать исполнение команды в Data View. По условию C0 выполнится команда CTRCLRT, которая очистит таблицу настроек для выхода 0 модуля CTRIO в слоте 2.

Порядок использования следующий: загрузить файл CTRCLRT_ex1.cwb на модуль CTRIO, затем ввести показанную здесь программу, перейти в режим RUN, в Data View включить выход 0 переводя бит C2 в значение On, переключить кодировщик на CTRIO на значение выше 10, и тогда индикатор выхода 0 загорится и будет гореть пока счет не достигнет 20, затем выключится. Теперь сбросьте счетчик с помощью C1, включите C0 чтобы выполнить команду CTRCLRT, которая очистит таблицу, включите выход 0 посредством C2, затем включите кодировщик на 10+ (выход 0 НЕ должен включиться).



Цепь 3: Эта цепь позволяет программисту сбросить счетчик из программы.



Цепь 4: Эта цепь позволяет оператору включить выход 0 из программы.



CTRIO Edit Preset Table Entry (CTREDPT) (IB-1003)

Для указанного выхода модуля CTRIO редактирует запись в таблице настроек модуля. Команда работает по переднему фронту. Эта команда удобна, когда необходимо отредактировать больше одной записи в таблице за один раз. Если вы хотите сделать только одну правку и затем сразу перезагрузить таблицу, то используйте CTRIO Edit and Reload Preset Table Entry (CTREDRL).

Команда требует для исполнения более одного программного цикла. По результату работы команды выставляется бит Success (успешное выполнение) или Error (ошибка при выполнении). Если возникла ошибка, используйте команду CTRIO Read Error Code (CTRRDER) для получения расширенной информации о ней.

Типы настроек (установок):

K0: Set – установить

K1: Reset – сбросить

K2: Pulse On – импульс On длительностью Pulse Time

K3: Pulse Off – импульс Off длительностью Pulse Time

K4: Toggle – переключить (инвертировать)

K5: Reset Count – сбросить счетчик

Заметьте. Параметр Pulse Time не используется некоторыми типами установок.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

Параметры:

- **CTRIO#:** логический уникальный номер модуля CTRIO, назначаемый пользователем.
- **Output#:** номер выхода модуля CTRIO.
- **Table#:** номер таблицы.
- **Entry#:** номер установки (записи).
- **Entry Type:** тип установки для добавления в таблицу.
- **Pulse Time:** время импульса для установок типа Pulse On and Pulse Off
- **Preset Count:** начальное значение счетчика, которое будет установлено после сброса.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
CTRIO# K	Константа: K0-255
Output# K	Константа: K0-3
Table# V,K	K0-255 и См. карту памяти DL06 - Слова данных
Entry# V,K	K0-255 и См. карту памяти DL06 - Слова данных
Entry Type V,K	K0-5 и См. карту памяти DL06 - Слова данных
Pulse Time V,K	K0-65535 и См. карту памяти DL06 - Слова данных
Preset Count V,K	K0-2147434528 и См. карту памяти DL06 - Слова данных
Workspace V	См. карту памяти DL06 - Слова данных

DirectSOFT5 Команды контроллера DL06

Success	X,Y,C,GX,GY,B	См. карту памяти DL06
Error	X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

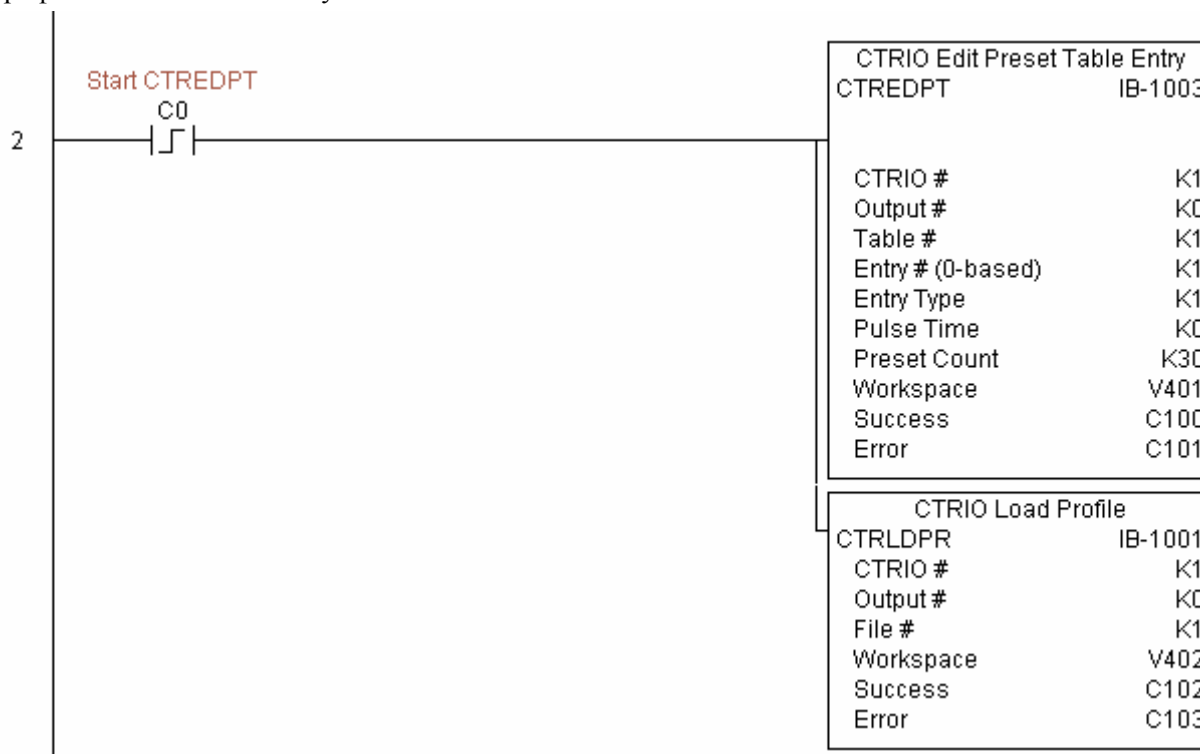
Цепь 1: Здесь привязывается модуль CTRIO к слоту 2 локального каркаса. Для каждого модуля CTRIO нужно использовать свою отдельную команду настройки CTRIO Config перед тем как использовать другие команды CTRxxxx, работающие с этим модулем. Модуль CTRIO конфигурируется на использование памяти V2000-V2025 для входных данных, и V2030-V2061 – для выходных данных.



Цепь 2: Это пример простого использования команды CTREDPT. Условие C здесь добавлено чтобы дать возможность программисту контролировать исполнение команды в Data View. Срабатывание условия выполнит команду CTREDPT, которая заменит вторую установку в таблице настроек для выхода 0 модуля CTRIO в слоте 2: со сброса по счету 20 на сброс по счету 30.

Порядок использования следующий: загрузить файл CTREDPT_ex1.cwb на модуль CTRIO, затем ввести показанную здесь программу, перейти в режим RUN, в Data View включить выход 0 переводя бит C2 в значение On, переключить кодировщик на CTRIO на значение выше 10, и тогда индикатор выхода 0 загорится и будет гореть пока счет не достигнет 20, затем выключится. Теперь сбросьте счетчик с помощью C1, включите C0 чтобы выполнить команду CTREDPT и изменить вторую настройку, включите выход 0 посредством C2, затем переведите кодировщик на 10+ (выход 0 должен включиться) и затем продолжайте до 30 и далее (выход 0 должен выключиться).

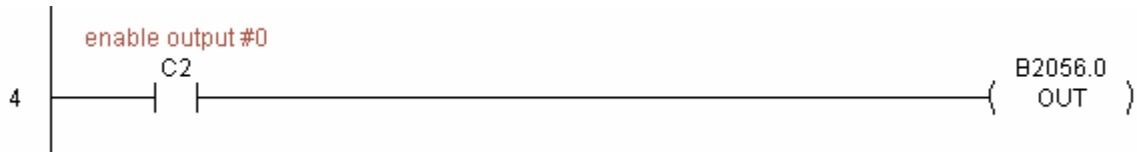
Здесь команда CTRLDPR следует сразу за командой CTREDPT, потому что мы должны перезагрузить профиль после изменения установок.



Цепь 3: Эта цепь позволяет программисту сбросить счетчик из программы.



Цепь 4: Эта цепь позволяет оператору включить выход 0 из программы.



CTRIO Edit Preset Table Entry and Reload (CTREDRL) (IB-1002)

Эта команда выполняет сразу правку записи в таблице настроек выхода модуля CTRIO и перезагрузку. Выполняется по переднему фронту.

Команда требует для исполнения более одного программного цикла. По результату работы команды выставляется бит Success (успешное выполнение) или Error (ошибка при выполнении). Если возникла ошибка, используйте команду CTRIO Read Error Code (CTRRDER) для получения расширенной информации о ней.

Типы настроек (установок):

K0: Set – установить

K1: Reset – сбросить

K2: Pulse On – импульс On длительностью Pulse Time

K3: Pulse Off – импульс Off длительностью Pulse Time

K4: Toggle – переключить (инвертировать)

K5: Reset Count – сбросить счетчик

Заметьте. Параметр Pulse Time не используется некоторыми типами установок.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

Параметры:

- **CTRIO#:** логический уникальный номер модуля CTRIO, назначаемый пользователем.
- **Output#:** номер выхода модуля CTRIO.
- **Table#:** номер таблицы.
- **Entry#:** номер установки (записи).
- **Entry Type:** тип установки для добавления в таблицу.
- **Pulse Time:** время импульса для установки типа Pulse On and Pulse Off
- **Preset Count:** начальное значение счетчика, которое будет установлено после сброса.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
CTRIO# K	Константа: K0-255
Output# K	Константа: K0-3
Table# V,K	K0-255 и См. карту памяти DL06 - Слова данных
Entry# V,K	K0-255 и См. карту памяти DL06 - Слова данных
Entry Type V,K	K0-5 и См. карту памяти DL06 - Слова данных
Pulse Time V,K	K0-65535 и См. карту памяти DL06 - Слова данных
Preset Count V,K	K0-2147434528 и См. карту памяти DL06 - Слова данных
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

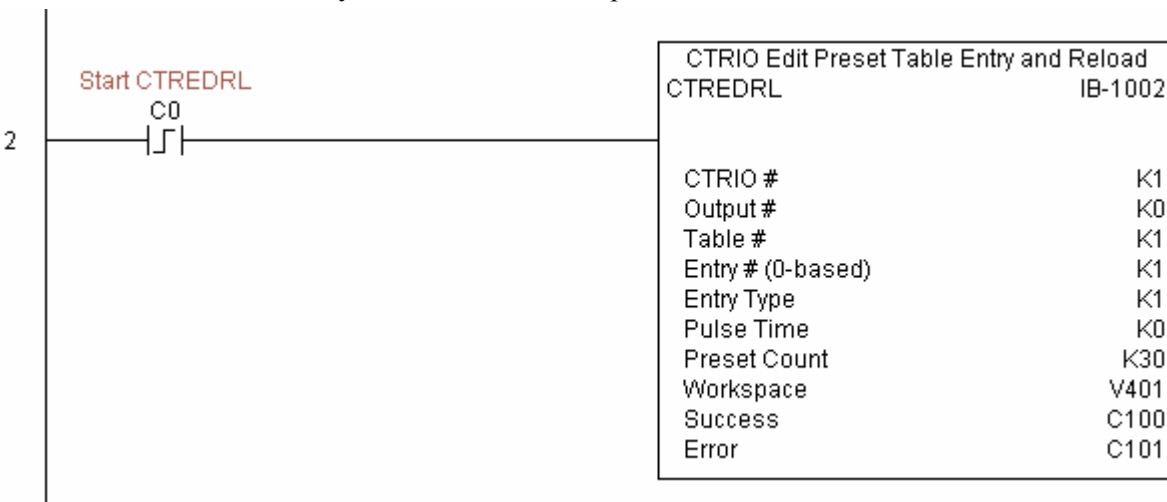
Цепь 1: Здесь привязывается модуль CTRIO к слоту 2 локального каркаса. Для каждого модуля CTRIO нужно использовать свою отдельную команду настройки CTRIO Config перед тем как использовать другие команды CTRxxxx, работающие с этим модулем. Модуль CTRIO конфигурируется на использование памяти V2000-V2025 для входных данных, и V2030-V2061 – для выходных данных.



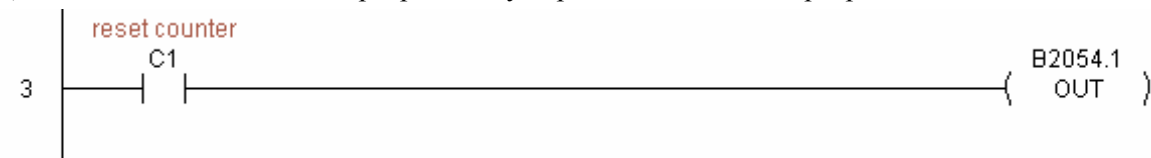
Цепь 2: Это пример простого использования команды CTREDRL. Условие C0 здесь добавлено чтобы дать возможность программисту контролировать исполнение команды в Data View. Срабатывание условия выполнит команду CTREDRL, которая изменит установку (preset) в таблице настроек 1 (file) со сброса по счету 20 на сброс по счету 30.

Порядок использования следующий: загрузить файл CTREDRL_ex1.cwb на модуль CTRIO, затем ввести показанную здесь программу, перейти в режим RUN, в Data View включить выход 0 переводя бит C2 в значение On, переключить кодировщик на CTRIO на значение выше 10, и тогда индикатор выхода 0 загорится, продолжайте счет выше 20, индикатор выхода 0 выключится. Теперь сбросьте счетчик с помощью C1, включите C0 чтобы выполнить команду CTREDRL и изменить вторую настройку на счет до 30, включите выход 0 посредством C2, затем переведите кодировщик на 10+ (выход 0 должен включиться) и затем продолжайте счет до 30 и далее (выход 0 должен выключиться).

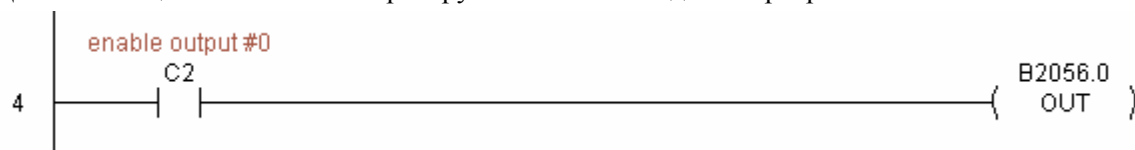
Обратите внимание, сейчас нет необходимости отдельно перегружать таблицу (file), однако, эта команда может изменять только одну запись в таблице за раз.



Цепь 3: Эта цепь позволяет программисту сбросить счетчик из программы.



Цепь 4: Эта цепь позволяет оператору включить выход 0 из программы.



CTRIO Initialize Preset Table (CTRINPT) (IB-1004)

По переднему фронту для одного выхода модуля CTRIO эта команда создает одну установку (запись) в таблице настроек в памяти, но не как файл.

Команда требует для исполнения более одного программного цикла. По результату работы команды выставляется бит Success (успешное выполнение) или Error (ошибка при выполнении). Если возникла ошибка, используйте команду CTRIO Read Error Code (CTRRDER) для получения расширенной информации о ней.

Типы настроек (установок):

K0: Set – установить

K1: Reset – сбросить

K2: Pulse On – импульс On длительностью Pulse Time

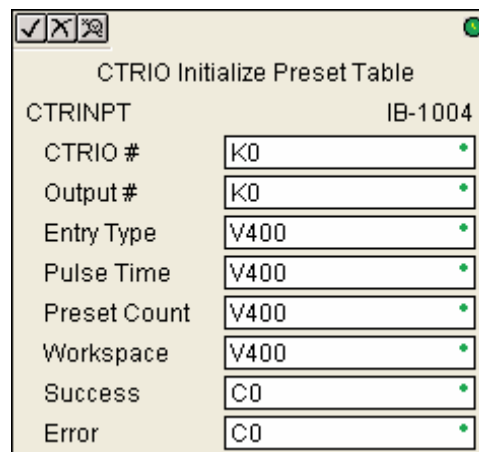
K3: Pulse Off – импульс Off длительностью Pulse Time

K4: Toggle – переключить (инвертировать)

K5: Reset Count – сбросить счетчик

Заметьте. Параметр Pulse Time не используется некоторыми типами установок.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.



Параметры:

- **CTRIO#:** логический уникальный номер модуля CTRIO, назначаемый пользователем.
- **Output#:** номер выхода модуля CTRIO.
- **Entry Type:** тип установки для добавления в таблицу.
- **Pulse Time:** время импульса для установок типа Pulse On and Pulse Off
- **Preset Count:** начальное значение счетчика, которое будет установлено после сброса.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
CTRIO# K	Константа: K0-255
Output# K	Константа: K0-3
Entry Type V,K	K0-5 и См. карту памяти DL06 - Слова данных
Pulse Time V,K	K0-65535 и См. карту памяти DL06 - Слова данных
Preset Count V,K	K0-2147434528 и См. карту памяти DL06 - Слова данных
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

Цепь 1: Здесь привязывается модуль CTRIO к слоту 2 локального каркаса. Для каждого модуля CTRIO нужно использовать свою отдельную команду настройки CTRIO Config перед тем как использовать другие команды CTRxxxx, работающие с этим модулем. Модуль CTRIO конфигурируется на использование памяти V2000-V2025 для входных данных, и V2030-V2061 – для выходных данных.

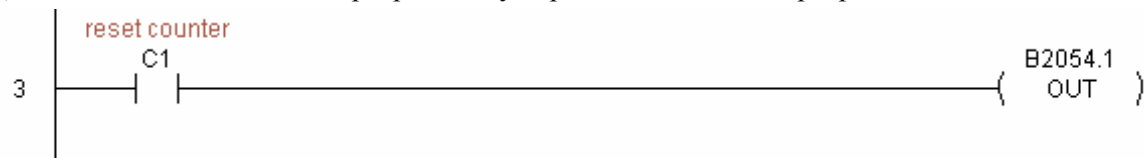


Цепь 2: Это пример простого использования команды CTRINPT. Условие C0 здесь добавлено чтобы дать возможность программисту контролировать исполнение команды в Data View. Срабатывание условия выполнит команду CTRINPT, которая создаст таблицу настроек с одной установкой (preset) для выхода 0, но не как файл (file). В этом случае в этой установке будет задан счетчик 15 для выхода 0.

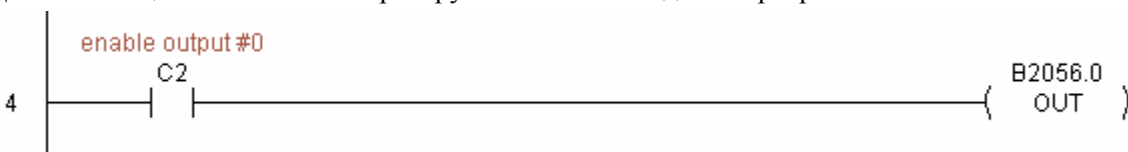
Порядок использования следующий: загрузить файл CTRINPT_ex1.cwb на модуль CTRIO, затем ввести показанную здесь программу, перейти в режим RUN, в Data View включить выход 0 переводя бит C2 в значение On, переключить кодировщик на CTRIO на значение выше 15, и индикатор выхода 0 не загорится. Теперь сбросьте счетчик с помощью C1, включите C0 чтобы выполнить команду CTRINPT и создать одну установку в таблице настроек со счетчиком 15 для выхода 0, затем переведите кодировщик на 15+ (выход 0 должен включиться).



Цепь 3: Эта цепь позволяет программисту сбросить счетчик из программы.



Цепь 4: Эта цепь позволяет оператору включить выход 0 из программы.



CTRIO Initialize Preset Table on Reset (CTRINTR) (IB-1010)

По каждому сбросу указанного выхода Nx-CTRIO эта команда создает новую таблицу настроек (Preset Table) для этого выхода. Новая таблица инициализируется одной установкой, сформированной из параметров этой команды.

Команда требует для исполнения более одного программного цикла. По результату работы команды выставляется бит Success (успешное выполнение) или Error (ошибка при выполнении). Если возникла ошибка, используйте команду CTRIO Read Error Code (CTRRDER) для получения расширенной информации о ней.

Типы настроек (установок):

K0: Set – установить

K1: Reset – сбросить

K2: Pulse On – импульс On длительностью Pulse Time

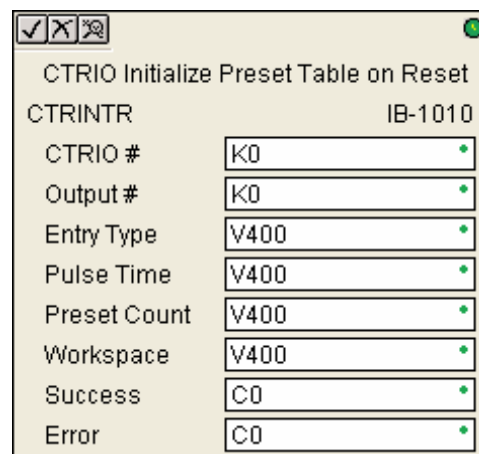
K3: Pulse Off – импульс Off длительностью Pulse Time

K4: Toggle – переключить (инвертировать)

K5: Reset Count – сбросить счетчик

Заметьте. Параметр Pulse Time не используется некоторыми типами установок.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.



Параметры:

- **CTRIO#:** логический уникальный номер модуля CTRIO, назначаемый пользователем.
- **Output#:** номер выхода Nx-CTRIO.
- **Entry Type:** тип установки для добавления в таблицу.
- **Pulse Time:** время импульса для установок типа Pulse On and Pulse Off
- **Preset Count:** начальное значение счетчика, которое будет установлено после сброса.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
CTRIO# K	Константа: K0-255
Output# K	Константа: K0-3
Entry Type V,K	K0-5 и См. карту памяти DL06 - Слова данных
Pulse Time V,K	K0-65535 и См. карту памяти DL06 - Слова данных
Preset Count V,K	K0-2147434528 и См. карту памяти DL06 - Слова данных
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

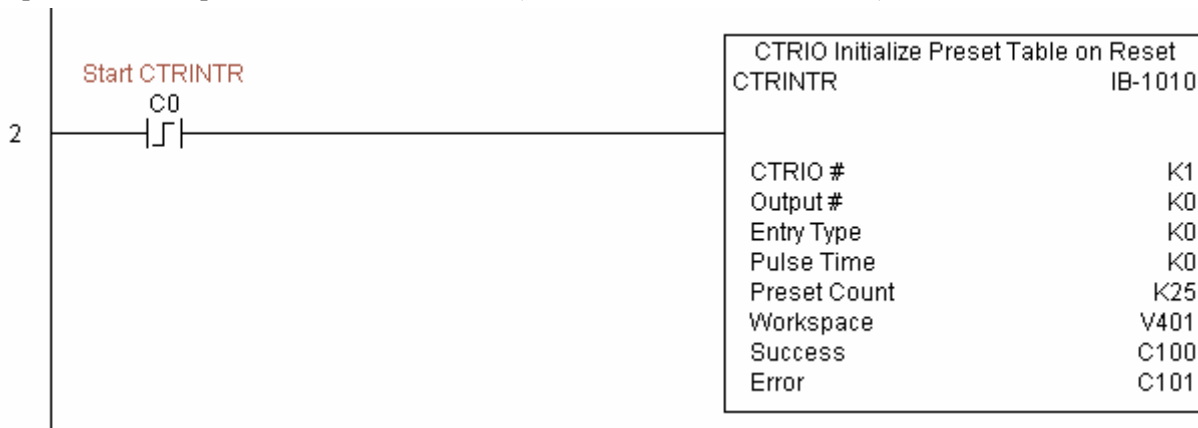
Пример:

Цепь 1: Здесь привязывается модуль CTRIO к слоту 2 локального каркаса. Для каждого модуля CTRIO нужно использовать свою отдельную команду настройки CTRIO Config перед тем как использовать другие команды CTRxxxx, работающие с этим модулем. Модуль CTRIO конфигурируется на использование памяти V2000-V2025 для входных данных, и V2030-V2061 – для выходных данных.

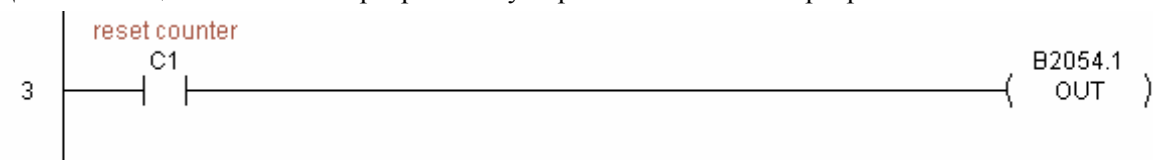


Цепь 2: Это пример простого использования команды CTRINTR. Условие C0 здесь добавлено чтобы дать возможность программисту контролировать исполнение команды в Data View. Срабатывание условия выполнит команду CTRINTR, которая создаст таблицу настроек с одной установкой (preset) для выхода 0, но не как файл (file). Новая установка будет загружена когда текущий счетчик будет сброшен. В этом случае в этой установке будет задан счетчик 25 для выхода 0.

Порядок использования следующий: загрузить файл CTRINTR_ex1.cwb на модуль CTRIO, затем ввести показанную здесь программу, перейти в режим RUN, в Data View включить выход 0 переводя бит C2 в значение On, переключить кодировщик на CTRIO на значение выше 10, и индикатор выхода 0 загорится. Теперь включите C0 чтобы выполнить команду CTRINTR, сбросьте счетчик с помощью C1, затем переведите кодировщик на значение 25+ (выход 0 должен включиться).



Цепь 3: Эта цепь позволяет программисту сбросить счетчик из программы.



Цепь 4: Эта цепь позволяет оператору включить выход 0 из программы.

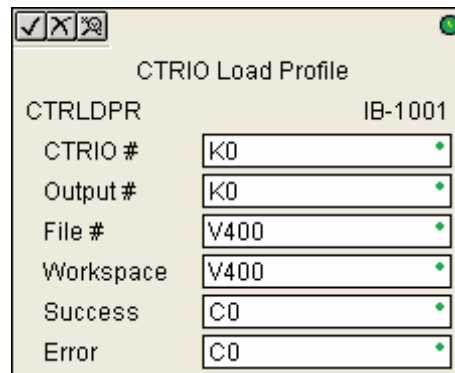


CTRIO Load Profile (CTRLDPR) (IB-1001)

По переднему фронту эта команда загружает файл профиля выхода модуля CTRIO.

Команда требует для исполнения более одного программного цикла. По результату работы команды выставляется бит Success (успешное выполнение) или Error (ошибка при выполнении). Если возникла ошибка, используйте команду CTRIO Read Error Code (CTRRDER) для получения расширенной информации о ней.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.



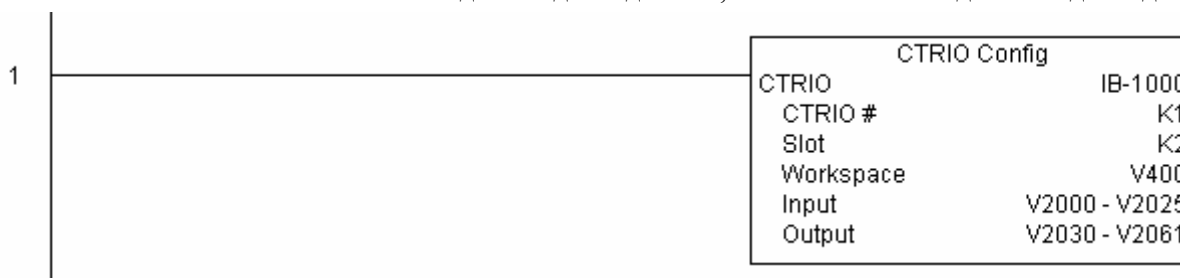
Параметры:

- **CTRIO#:** логический уникальный номер модуля CTRIO, назначаемый пользователем.
- **Output#:** номер выхода модуля CTRIO.
- **File:** номер файла профиля, который должен быть загружен.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
CTRIO# K	Константа: K0-255
Output# K	Константа: K0-3
File# V,K	K0-255 и См. карту памяти DL06 - Слова данных
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

Цепь 1: Здесь привязывается модуль CTRIO к слоту 2 локального каркаса. Для каждого модуля CTRIO нужно использовать свою отдельную команду настройки CTRIO Config перед тем как использовать другие команды CTRxxxx, работающие с этим модулем. Модуль CTRIO конфигурируется на использование памяти V2000-V2025 для входных данных, и V2030-V2061 – для выходных данных.



Цепь 2: Эта команда загружает профиль №1 в рабочую память для выхода 0 модуля CTRIO с номером 1. Для этого примера необходимо чтобы был загружен CTRLDPR_IB0x.cwb в ваш Nx-CTRIO модуль.



Цепь 3: Если файл успешно загружен, устанавливается Profile_Loaded.



CTRIO Read Error (CTRRDER) (IB-1014)

По переднему фронту эта команда получает от модуля CTRIO десятичный код ошибки и помещает его в указываемый регистр памяти (параметр Error Code).

Поскольку код ошибки хранится неизменным пока не выполнится следующая команда на модуль CTRIO, то нужно использовать этот код сразу после команды, которая сообщала об ошибке своим битом Error.

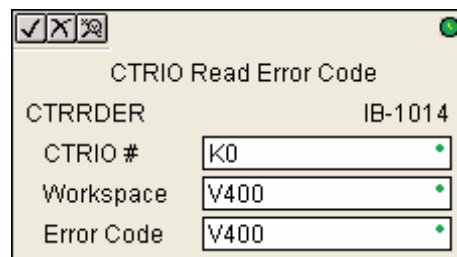
Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

Коды ошибок:

- 0: нет ошибок
- 100: указанный код команды неизвестен или не поддерживается
- 101: файл с таким номером не найден в файловой системе
- 102: тип файла не верен для указанной выходной функции
- 103: файл профиля не известен
- 104: указанный вход не настроен как лимит на этом входе
- 105: указанный ограничительный входной порог вне допустимых пределов
- 106: указанная входная функция не сконфигурирована либо неверна
- 107: указанный номер входной функции вне допустимых пределов
- 108: указанная функция настройки неверна
- 109: таблица настроек полна
- 110: указанная запись таблицы вне допустимых пределов
- 111: указанный номер регистра вне допустимых пределов
- 112: указанный регистр не настроен как вход или выход
- 2001: невозможно получить код ошибки – нет доступа к модулю CTRIO через ERM

Параметры:

- **CTRIO#**: логический уникальный номер модуля CTRIO, назначаемый пользователем.
- **Workspace**: адрес ячейки памяти, который будет использован командой.
- **Error Code**: адрес ячейки памяти, куда будет записан код ошибки.



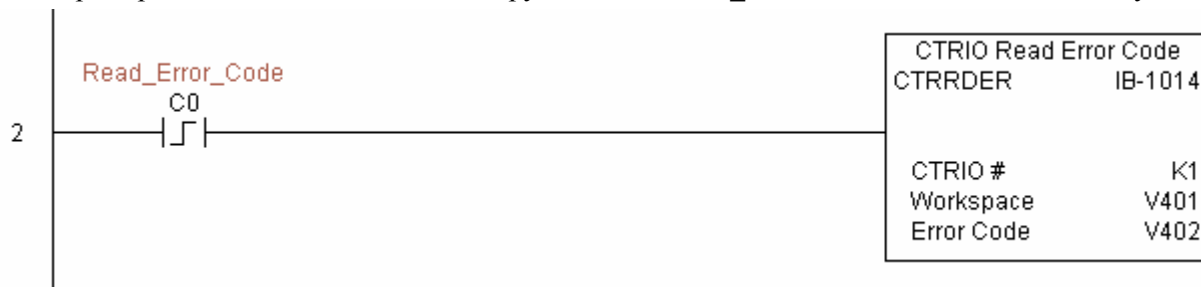
Параметр	Диапазон DL06
CTRIO# K	Константа: K0-255
Workspace V	См. карту памяти DL06 - Слова данных
Error Code V	См. карту памяти DL06 - Слова данных

Пример:

Цепь 1: Здесь привязывается модуль CTRIO к слоту 2 локального каркаса. Для каждого модуля CTRIO нужно использовать свою отдельную команду настройки CTRIO Config перед тем как использовать другие команды CTRxxxx, работающие с этим модулем. Модуль CTRIO конфигурируется на использование памяти V2000-V2025 для входных данных, и V2030-V2061 – для выходных данных.



Цепь 2: Эта команда читает расширенную информацию об ошибке из модуля CTRIO с номером 1. Для этого примера необходимо чтобы был загружен CTRRDER_IBox.cwb в ваш Hx-CTRIO модуль.



CTRIO Run to Limit Mode (CTRRTLМ) (IB-1011)

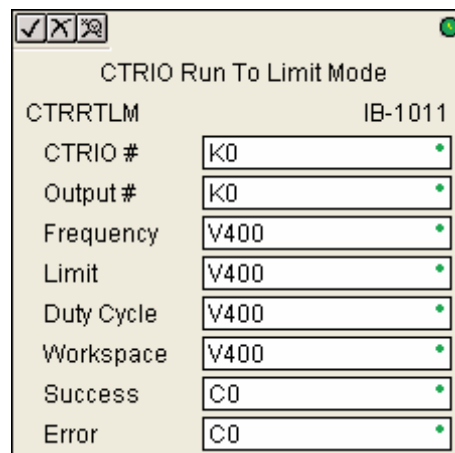
По переднему фронту эта инструкция загружает команду Run to Limit и заданные параметры на указанный выход модуля CTRIO. Для работы этой инструкции входы CTRIO должны быть настроены как Limits.

Допустимые шестнадцатеричные значения пределов:

- K00 – передний фронт Ch1/C
- K10 – задний фронт Ch1/C
- K20 – оба фронта Ch1/C
- K01 – передний фронт Ch1/D
- K11 – задний фронт Ch1/D
- K21 – оба фронта Ch1/D
- K02 – передний фронт Ch2/C
- K12 – задний фронт Ch2/C
- K22 – оба фронта Ch2/C
- K03 – передний фронт Ch2/D
- K13 – задний фронт Ch2/D
- K23 – оба фронта Ch2/D

Команда требует для исполнения более одного программного цикла. По результату работы команды выставляется бит Success (успешное выполнение) или Error (ошибка при выполнении). Если возникла ошибка, используйте команду CTRIO Read Error Code (CTRORDER) для получения расширенной информации о ней.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.



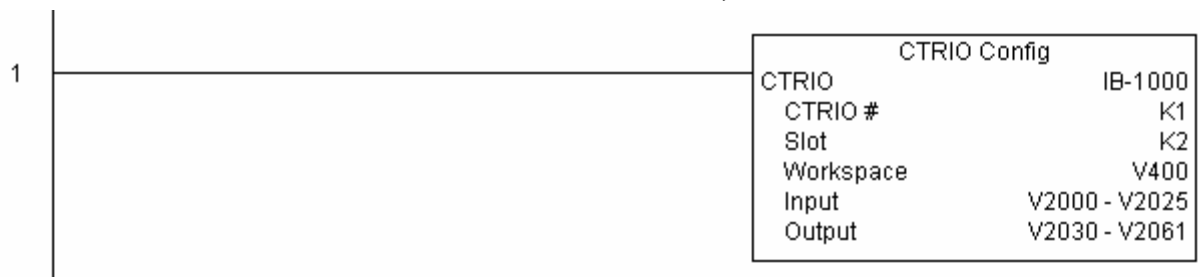
Параметры:

- **CTRIO#:** логический уникальный номер модуля CTRIO, назначаемый пользователем.
- **Output#:** номер выхода модуля CTRIO.
- **Frequency:** выходная частота импульсов (20Гц - 20КГц)
- **Limit:** входы (Input) CTRIO должны быть настроены как пределы (Limits)
- **Duty Cycle:** скважность импульсов, в процентах. Указывается в шестнадцатеричной форме. По умолчанию задан 0, что есть 50%, ввод 50 также соответствует 50%.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
CTRIO# K	Константа: K0-255
Output# K	Константа: K0-3
Frequency V,K	K20-20000 и См. карту памяти DL06 - Слова данных
Limit V,K	K0-FF и См. карту памяти DL06 - Слова данных
Duty Cycle V,K	K0-99 и См. карту памяти DL06 - Слова данных
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

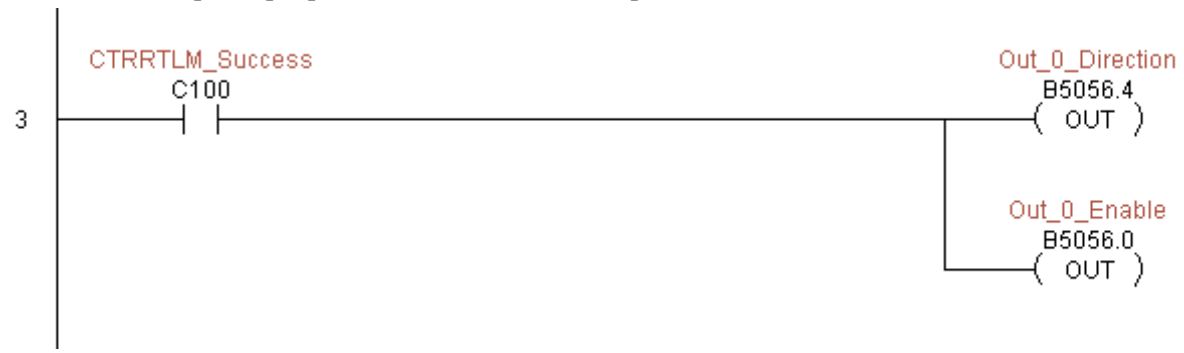
Цепь 1: Здесь привязывается модуль CTRIO к слоту 2 локального каркаса. Для каждого модуля CTRIO нужно использовать свою отдельную команду настройки CTRIO Config перед тем как использовать другие команды CTRxxxx, работающие с этим модулем. Модуль CTRIO конфигурируется на использование памяти V2000-V2025 для входных данных, и V2030-V2061 – для выходных данных.



Цепь 2: Эта команда задает на выход 0 модуля CTRIO с номером 1 выходные импульсы с частотой 1000 Гц пока не сработает вход 0. Для этого примера необходимо чтобы был загружен CTRRTLM_IBox.cwb в ваш Hx-CTRIO модуль.



Цепь 3: Если параметры режима Run To Limit в порядке, то выставляются биты Direction и Enable.



CTRIO Run to Position Mode (CTRRTPM) (IB-1012)

По переднему фронту эта инструкция загружает команду Run to Position и заданные параметры на указанный выход модуля CTRIO.

Допустимые значения функции:

- 00 – меньше чем Ch1/Fn1
- 10 – больше чем Ch1/Fn1
- 01 – меньше чем Ch1/Fn2
- 11 – больше чем Ch1/Fn2
- 02 – меньше чем Ch2/Fn1
- 12 – больше чем Ch2/Fn1
- 03 – меньше чем Ch2/Fn2
- 13 – больше чем Ch2/Fn2

Команда требует для исполнения более одного программного цикла.

По результату работы команды выставляется бит Success (успешное выполнение) или Error (ошибка при выполнении). Если возникла ошибка, используйте команду CTRIO Read Error Code (CTRRDER) для получения расширенной информации о ней.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.

Параметры:

- **CTRIO#:** логический уникальный номер модуля CTRIO, назначаемый пользователем.
- **Output#:** номер выхода модуля CTRIO.
- **Frequency:** выходная частота импульсов (20Гц - 20КГц)
- **Duty Cycle:** скважность импульсов, в процентах. Указывается в шестнадцатеричной форме. По умолчанию задан 0, что есть 50%, ввод 50 также соответствует 50%.
- **Position:** значение счетчика входных импульсов, по достижению которого последовательность выходных импульсов отключится.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
CTRIO# K	Константа: K0-255
Output# K	Константа: K0-3
Frequency V,K	K0-20000 и См. карту памяти DL06 - Слова данных
Position V,K	K0-2147434528 и См. карту памяти DL06 - Слова данных
Duty Cycle V,K	K0-99 и См. карту памяти DL06 - Слова данных
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

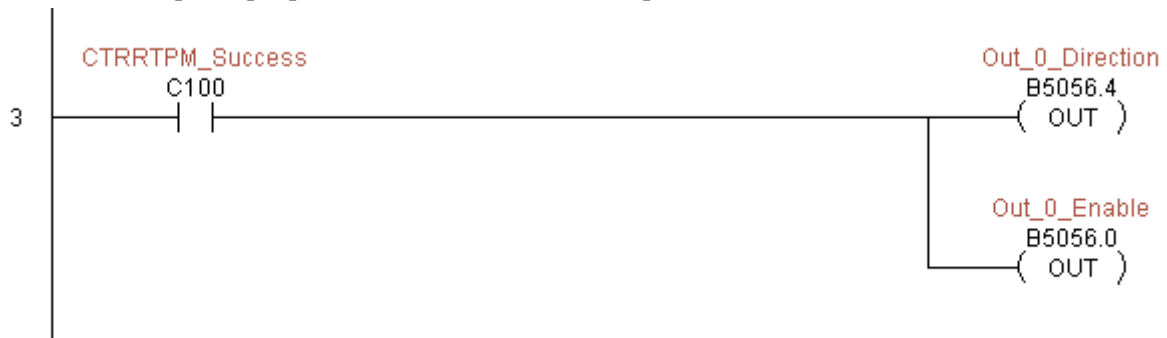
Цепь 1: Здесь привязывается модуль CTRIO к слоту 2 локального каркаса. Для каждого модуля CTRIO нужно использовать свою отдельную команду настройки CTRIO Config перед тем как использовать другие команды CTRxxxx, работающие с этим модулем. Модуль CTRIO конфигурируется на использование памяти V2000-V2025 для входных данных, и V2030-V2061 – для выходных данных.



Цепь 2: Эта команда задает на выход 0 модуля CTRIO с номером 1 выходные импульсы с частотой 1000 Гц, и по результату сравнения «больше чем Ch1/Fn1», при достижении на счетном входе 1500, выходные импульсы прекращаются. Для этого примера необходимо чтобы был загружен CTRRTPM_IBox.cwb в ваш Hx-CTRIO модуль.



Цепь 3: Если параметры режима Run To Position в порядке, то выставляются биты Direction и Enable.

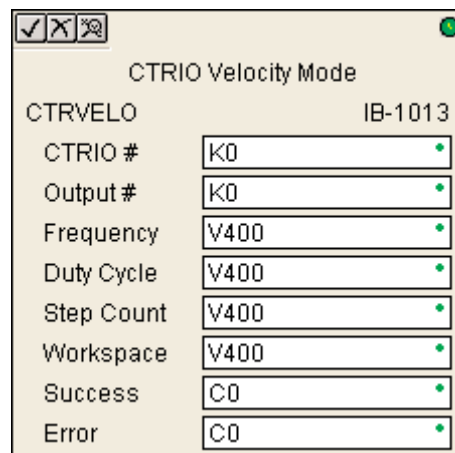


CTRIO Velocity Mode (CTRVELO) (IB-1013)

По переднему фронту эта инструкция выдает команду Velocity и заданные параметры на указанный выход модуля CTRIO.

Команда требует для исполнения более одного программного цикла. По результату работы команды выставляется бит Success (успешное выполнение) или Error (ошибка при выполнении). Если возникла ошибка, используйте команду CTRIO Read Error Code (CTRRDER) для получения расширенной информации о ней.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.



Параметры:

- **CTRIO#:** логический уникальный номер модуля CTRIO, назначаемый пользователем.
- **Output#:** номер выхода модуля CTRIO.
- **Frequency:** выходная частота импульсов (20Гц - 20КГц)
- **Duty Cycle:** скважность импульсов, в процентах. Указывается в шестнадцатеричной форме. По умолчанию задан 0, что есть 50%, ввод 50 также соответствует 50%.
- **Step Count:** количество выходных импульсов в виде 32-битного шестнадцатеричного числа. Значение Kfffffff задаст непрерывную последовательность импульсов пока выход включен.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

Параметр	Диапазон DL06
CTRIO# K	Константа: K0-255
Output# K	Константа: K0-3
Frequency V,K	K0-20000 и См. карту памяти DL06 - Слова данных
Step Count V,K	K0-2147434528 и См. карту памяти DL06 - Слова данных
Duty Cycle V,K	K0-99 и См. карту памяти DL06 - Слова данных
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

Цепь 1: Здесь привязывается модуль CTRIO к слоту 2 локального каркаса. Для каждого модуля CTRIO нужно использовать свою отдельную команду настройки CTRIO Config перед тем как использовать другие команды CTRxxxx, работающие с этим модулем. Модуль CTRIO конфигурируется на использование памяти V2000-V2025 для входных данных, и V2030-V2061 – для выходных данных.

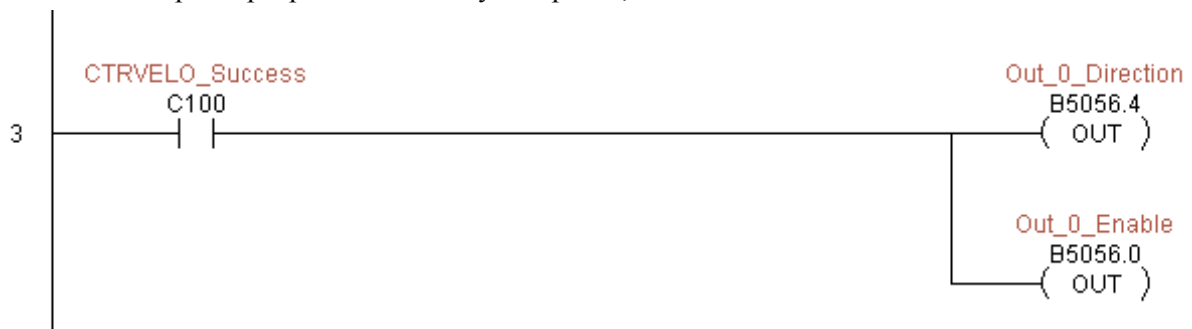


DirectSOFT5 Команды контроллера DL06

Цепь 2: Эта команда задает на выход 0 модуля CTRIO с номером 1 выходные импульсы с частотой 1000 Гц в количестве 10000. Для этого примера необходимо чтобы был загружен CTRVELO_IBox.cwb в ваш Hx-CTRIO модуль.



Цепь 3: Если параметры режима Velocity в порядке, то выставляются биты Direction и Enable.

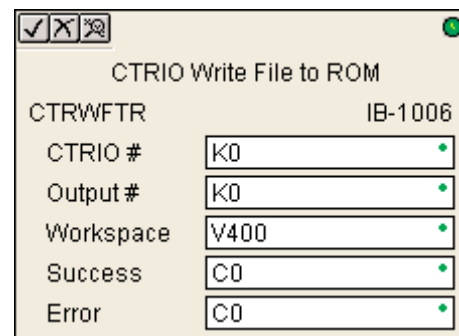


CTRIO Write File to ROM (CTRWFTR) (IB-1006)

Эта команда записывает измененную в памяти таблицу настроек модуля CTRIO назад во Flash ROM память. Команда работает по переднему фронту.

Команда требует для исполнения более одного программного цикла. По результату работы команды выставляется бит Success (успешное выполнение) или Error (ошибка при выполнении). Если возникла ошибка, используйте команду CTRIO Read Error Code (CTRRDER) для получения расширенной информации о ней.

Ячейка памяти, задаваемая параметром WORKSPACE, привязывается только в этой команде, и НЕ ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ ГДЕ-ЛИБО в программе еще.



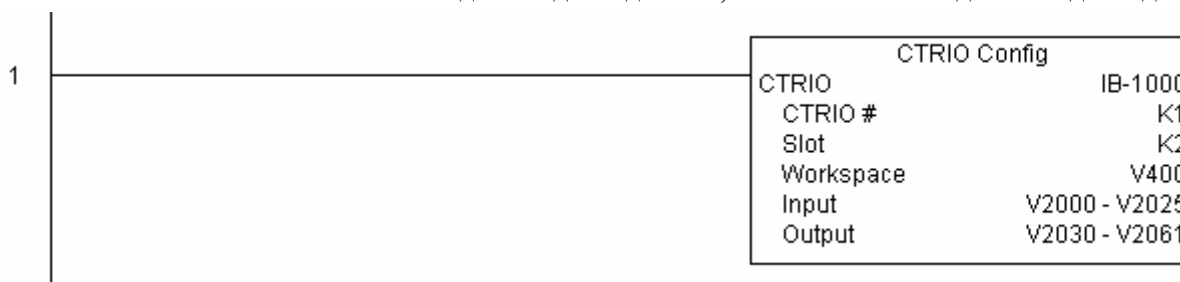
Параметры:

- **CTRIO#:** логический уникальный номер модуля CTRIO, назначаемый пользователем.
- **Output#:** номер выхода модуля CTRIO.
- **Workspace:** адрес ячейки памяти, который будет использован командой.
- **Success:** адрес бита, который будет установлен при успешном выполнении команды.
- **Error:** адрес бита, который будет установлен при неудачном выполнении команды.

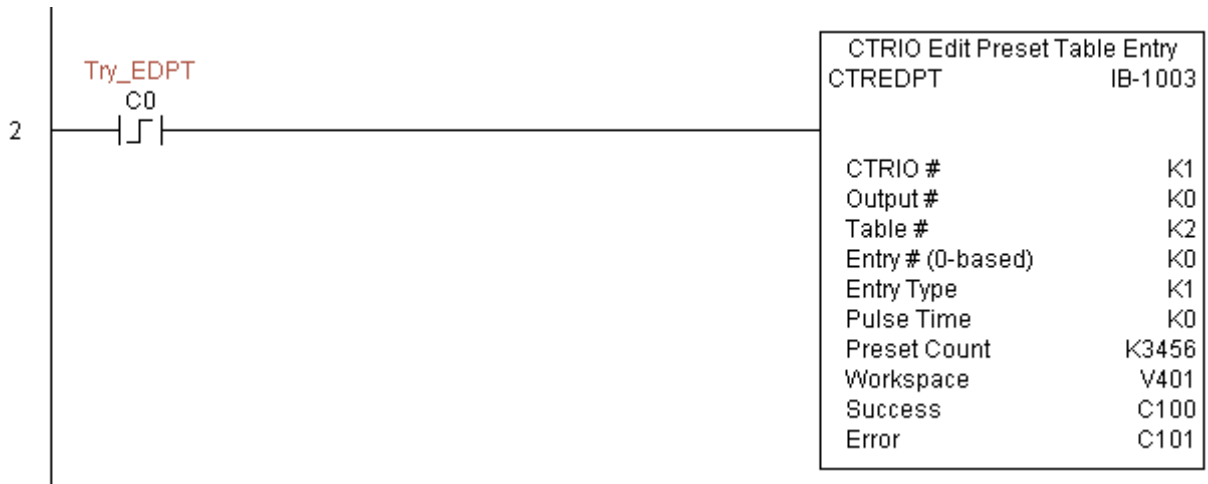
Параметр	Диапазон DL06
CTRIO# K	Константа: K0-255
Output# K	Константа: K0-3
Workspace V	См. карту памяти DL06 - Слова данных
Success X,Y,C,GX,GY,B	См. карту памяти DL06
Error X,Y,C,GX,GY,B	См. карту памяти DL06

Пример:

Цепь 1: Здесь привязывается модуль CTRIO к слоту 2 локального каркаса. Для каждого модуля CTRIO нужно использовать свою отдельную команду настройки CTRIO Config перед тем как использовать другие команды CTRxxxx, работающие с этим модулем. Модуль CTRIO конфигурируется на использование памяти V2000-V2025 для входных данных, и V2030-V2061 – для выходных данных.



Цепь 2: Эта команда изменяет запись (установку) 0 в таблице настроек 2, устанавливая сброс по счету 3456. Для этого примера необходимо чтобы был загружен CTRWFTR_IBox.cwb в ваш Hx-CTRIO модуль.



Цепь 3: Если файл успешно изменен, используйте команду Write File To ROM для сохранения таблицы во Flash ROM памяти модуля.

