DL405 IBox Instructions
PLC User Manual Supplement

Manual Number: DL405-IBOX-S

⚡ WARNING ⚡

Thank you for purchasing automation equipment from Automationdirect.com™, doing business as AutomationDirect. We want your new automation equipment to operate safely. Anyone who installs or uses this equipment should read this publication (and any other relevant publications) before installing or operating the equipment.

To minimize the risk of potential safety problems, you should follow all applicable local and national codes that regulate the installation and operation of your equipment. These codes vary from area to area and usually change with time. It is your responsibility to determine which codes should be followed, and to verify that the equipment, installation, and operation is in compliance with the latest revision of these codes.

At a minimum, you should follow all applicable sections of the National Fire Code, National Electrical Code, and the codes of the National Electrical Manufacturer's Association (NEMA). There may be local regulatory or government offices that can also help determine which codes and standards are necessary for safe installation and operation.

Equipment damage or serious injury to personnel can result from the failure to follow all applicable codes and standards. We do not guarantee the products described in this publication are suitable for your particular application, nor do we assume any responsibility for your product design, installation, or operation.

Our products are not fault-tolerant and are not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the product could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). AutomationDirect specifically disclaims any expressed or implied warranty of fitness for High Risk Activities.

For additional warranty and safety information, see the Terms and Conditions section of our catalog. If you have any questions concerning the installation or operation of this equipment, or if you need additional information, please call us at 770-844-4200.

This publication is based on information that was available at the time it was printed. At AutomationDirect we constantly strive to improve our products and services, so we reserve the right to make changes to the products and/or publications at any time without notice and without any obligation. This publication may also discuss features that may not be available in certain revisions of the product.

Trademarks

This publication may contain references to products produced and/or offered by other companies. The product and company names may be trademarked and are the sole property of their respective owners. AutomationDirect disclaims any proprietary interest in the marks and names of others.

AVERTISSEMENT

Nous vous remercions d'avoir acheté l'équipement d'automatisation de Automationdirect.com™, en faisant des affaires comme AutomationDirect. Nous tenons à ce que votre nouvel équipement d'automatisation fonctionne en toute sécurité. Toute personne qui installe ou utilise cet équipement doit lire la présente publication (et toutes les autres publications pertinentes) avant de l'installer ou de l'utiliser.

Afin de réduire au minimum le risque d'éventuels problèmes de sécurité, vous devez respecter tous les codes locaux et nationaux applicables régissant l'installation et le fonctionnement de votre équipement. Ces codes diffèrent d'une région à l'autre et, habituellement, évoluent au fil du temps. Il vous incombe de déterminer les codes à respecter et de vous assurer que l'équipement, l'installation et le fonctionnement sont conformes aux exigences de la version la plus récente de ces codes.

Vous devez, à tout le moins, respecter toutes les sections applicables du Code national de prévention des incendies, du Code national de l'électricité et des codes de la National Electrical Manufacturer's Association (NEMA). Des organismes de réglementation ou des services gouvernementaux locaux peuvent également vous aider à déterminer les codes ainsi que les normes à respecter pour assurer une installation et un fonctionnement sûrs.

L'omission de respecter la totalité des codes et des normes applicables peut entraîner des dommages à l'équipement ou causer de graves blessures au personnel. Nous ne garantissons pas que les produits décrits dans cette publication conviennent à votre application particulière et nous n'assumons aucune responsabilité à l'égard de la conception, de l'installation ou du fonctionnement de votre produit.

Nos produits ne sont pas insensibles aux défaillances et ne sont ni conçus ni fabriqués pour l'utilisation ou la revente en tant qu'équipement de commande en ligne dans des environnements dangereux nécessitant une sécurité absolue, par exemple, l'exploitation d'installations nucléaires, les systèmes de navigation aérienne ou de communication, le contrôle de la circulation aérienne, les équipements de survie ou les systèmes d'armes, pour lesquels la défaillance du produit peut provoquer la mort, des blessures corporelles ou de graves dommages matériels ou environnementaux («activités à risque élevé»). La société AutomationDirect nie toute garantie expresse ou implicite d'aptitude à l'emploi en ce qui a trait aux activités à risque élevé.

Pour des renseignements additionnels touchant la garantie et la sécurité, veuillez consulter la section Modalités et conditions de notre documentation. Si vous avez des questions au sujet de l'installation ou du fonctionnement de cet équipement, ou encore si vous avez besoin de renseignements supplémentaires, n'hésitez pas à nous téléphoner au 770-844-4200.

Cette publication s'appuie sur l'information qui était disponible au moment de l'impression. À la société AutomationDirect, nous nous efforçons constamment d'améliorer nos produits et services. C'est pourquoi nous nous réservons le droit d'apporter des modifications aux produits ou aux publications en tout temps, sans préavis ni quelque obligation que ce soit. La présente publication peut aussi porter sur des caractéristiques susceptibles de ne pas être offertes dans certaines versions révisées du produit.

Marques de commerce

La présente publication peut contenir des références à des produits fabriqués ou offerts par d'autres entreprises. Les désignations des produits et des entreprises peuvent être des marques de commerce et appartiennent exclusivement à leurs propriétaires respectifs. AutomationDirect nie tout intérêt dans les autres marques et désignations.

Notes

AUTOMATIONDIRECT.com

Please include the Manual Number and the Manual Issue, both shown below, when communicating with Technical Support regarding this publication.

Manual Number:	**DL405-IBOX-S (Supplement to D4-USER-M)**
Issue:	**Original Edition, Rev B**
Issue Date:	**7/14**

Publication History

Issue	Date	Description of Changes
1st Edition	3/06	Original Edition
Rev. A	2/11	Minor Corrections
Rev. B	7/14	Added ***Direct***SOFT6 IBox Instructions

Overview

The IBox Instructions listed in this supplement are in addition to the Standard RLL Instructions found in Chapter 5 of the DL405 User Manual. This supplement contains IBox instructions that are available in *Direct*SOFT5 and those that are available in *Direct*SOFT6. The new IBox instructions for *Direct*SOFT6 are labelled as *Direct*SOFT6 ONLY.

For more information on *Direct*SOFT and to download our Free version, please visit our Web site at: www.automationdirect.com

Analog Helper IBoxes		
Instruction	**Ibox #**	**Page**
Analog Scale 12 Bit BCD to BCD (ANSCL)	IB-423	10
Analog Scale 12 Bit Binary to Binary (ANSCLB)	IB-403	12
Filter Over Time - BCD (FILTER)	IB-422	14
Filter Over Time - Binary (FILTERB)	IB-402	16
Hi/Low Alarm - BCD (HILOAL)	IB-421	18
Hi/Low Alarm - Binary (HILOALB)	IB-401	20

Discrete Helper IBoxes		
Instruction	**Ibox #**	**Page**
Off Delay Timer (OFFDTMR)	IB-302	22
On Delay Timer (ONDTMR)	IB-301	24
One Shot (ONESHOT)	IB-303	26
Push On / Push Off Circuit (PONOFF)	IB-300	28

Memory IBoxes		
Instruction	**Ibox #**	**Page**
Move Single Word (MOVEW)	IB-200	30
Move Double Word (MOVED)	IB-201	32

Math IBoxes		
Instruction	**Ibox #**	**Page**
BCD to Real with Implied Decimal Point (BCDTOR)	IB-560	34
Double BCD to Real with Implied Decimal Point (BCDTORD)	IB-562	36
Math - BCD (MATHBCD)	IB-521	38
Math - Binary (MATHBIN)	IB-501	40
Math - Real (MATHR)	IB-541	42
Real to BCD with Implied Decimal Point and Rounding (RTOBCD)	IB-561	44
Real to Double BCD with Implied Decimal Point and Rounding (RTOBCDD)	IB-563	46
Square BCD (SQUARE)	IB-523	48
Square Binary (SQUAREB)	IB-503	50
Square Real(SQUARER)	IB-543	52
Sum BCD Numbers (SUMBCD)	IB-522	54
Sum Binary Numbers (SUMBIN)	IB-502	56
Sum Real Numbers (SUMR)	IB-542	58

New IBox Instructions with *Direct*SOFT6

The following IBox instructions are only available with *Direct*SOFT6.

Analog Helper IBoxes		
Instruction	**Ibox #**	**Page**
Filter Over Time - BCD Double (FILTERD)	IB-425	158
Hi/Lo Alarm - Binary Double (HILOALBD)	IB-404	160
Hi/Lo Alarm - BCD Double (HILOALD)	IB-424	162

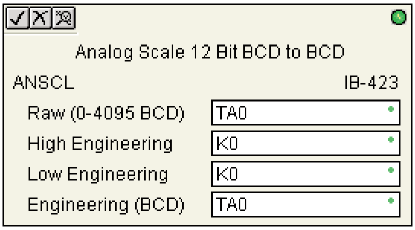
Memory IBoxes		
Instruction	**Ibox #**	**Page**
Move Real (MOVER)	IB-202	164
Move Range of V Using MOV (MOVRANGE)	IB-203	166
Move Range of V Using FOR/NEXT (MOVEFOR)	IB-204	168

Math IBoxes		
Instruction	**Ibox #**	**Page**
Absolute Value - Binary (ABSBIN)	IB-504	170
Unsigned Binary to Real with Implied Decimal Point (BINTOR)	IB-564	172
Signed Binary to Real with Implied Decimal Point (BINSTOR)	IB-568	174
Unsigned Double Binary to Real with Implied Decimal Point (BINTORD)	IB-566	176
Signed Double Binary to Real with Implied Decimal Point (BINSTORD)	IB-570	178
Real to Unsigned Binary with Implied Decimal Point and Rounding (RTOBIN)	IB-565	180
Real to Double Unsigned Binary with Implied Decimal Point and Rounding (RTOBIND)	IB-567	182
Real to Signed Binary with Implied Decimal Point and Rounding (RTOBINS)	IB-569	184
Real to Double Signed Binary with Implied Decimal Point and Rounding (RTOBINSD)	IB-571	186
Scale Value - Unsigned Binary (SCALEB)	IB-509	188
Decrement By Binary (DECBYBIN)	IB-507	190
Decrement By Binary Double (DECBYBIND)	IB-508	192
Decrement By BCD (DECBYBCD)	IB-526	194
Decrement By BCD Double (DECBYBCDD)	IB-527	196
Decrement By Real (DECBYR)	IB-546	198
Increment By Binary (INCBYBIN)	IB-505	200
Increment By Binary Double (INCBYBIND)	IB-506	202
Increment By BCD (INCBYBCD)	IB-524	204
Increment By BCD Double (INCBYBCDD)	IB-525	206
Increment By Real (INCBYR)	IB-545	208

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

DS5/6	Used
HPP	N/A

Analog Scale 12 Bit BCD to BCD scales a 12 bit BCD analog value (0-4095 BCD) into BCD engineering units. You specify the engineering unit high value (when raw is 4095), and the engineering low value (when raw is 0), and the output V memory address you want the to place the scaled engineering unit value. The engineering units are generated as BCD and can be the full range of 0 to 9999 (see ANSCLB - Analog Scale 12 Bit Binary to Binary if your raw units are in Binary format).



Analog Scale 12 Bit BCD to BCD

ANSCL IB-423

Raw (0-4095 BCD) TA0

High Engineering K0

Low Engineering K0

Engineering (BCD) TA0

Note that this IBox only works with unipolar unsigned raw values. It does NOT work with bipolar or sign plus magnitude raw values.

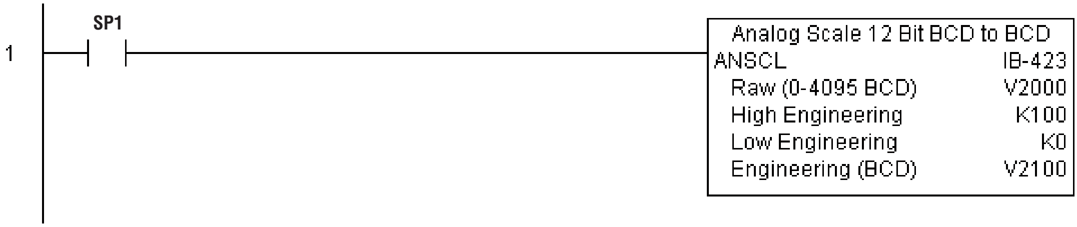
ANSCL Parameters

- Raw (0-4095 BCD): specifies the V-memory location of the unipolar unsigned raw 0-4095 unscaled value
- High Engineering: specifies the high engineering value when the raw input is 4095
- Low Engineering: specifies the low engineering value when the raw input is 0
- Engineering (BCD): specifies the V-memory location where the scaled engineering BCD value will be placed

Parameter	DL405 Range
Raw (0-4095 BCD) . . . . V,P	See DL405 V-memory map - Data Words
High Engineering . . . . K	K0-9999
Low Engineering . . . . K	K0-9999
Engineering (BCD). . . . . V,P	See DL405 V-memory map - Data Words

ANSCL Example

In the following example, the ANSCL instruction is used to scale a raw value (0-4095 BCD) that is in V2000. The engineering scaling range is set 0-100 (low engineering value - high engineering value). The scaled value will be placed in V2100 in BCD format.



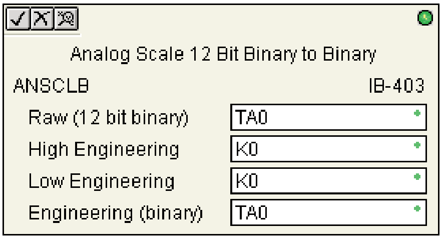
Analog Scale 12 Bit BCD to BCD

ANSCL	IB-423
Raw (0-4095 BCD)	V2000
High Engineering	K100
Low Engineering	K0
Engineering (BCD)	V2100

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

DS5/6	Used
HPP	N/A

Analog Scale 12 Bit Binary to Binary scales a 12 bit binary analog value (0-4095 decimal) into binary (decimal) engineering units. You specify the engineering unit high value (when raw is 4095), and the engineering low value (when raw is 0), and the output V memory address you want to place the scaled engineering unit value. The engineering units are generated as binary and can be the full range of 0 to 65535 (see ANSCL - Analog Scale 12 Bit BCD to BCD if your raw units are in BCD format).


Analog Scale 12 Bit Binary to Binary

ANSCLB IB-403

Raw (12 bit binary) TA0

High Engineering K0

Low Engineering K0

Engineering (binary) TA0

Note that this IBox only works with unipolar unsigned raw values. It does NOT work with bipolar, sign plus magnitude, or signed 2's complement raw values.

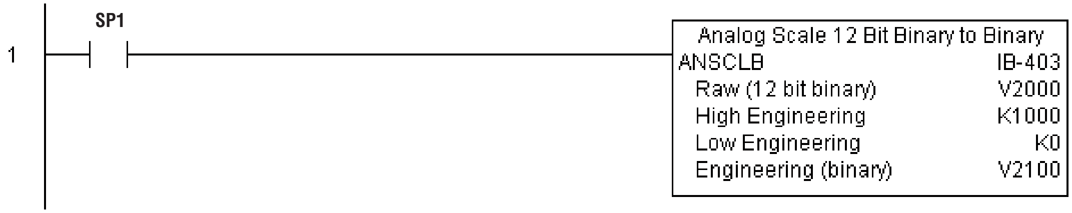
ANSCLB Parameters

- Raw (12 bit binary): specifies the V-memory location of the unipolar unsigned raw decimal unscaled value (12 bit binary = 0-4095 decimal)
- High Engineering: specifies the high engineering value when the raw input is 4095 decimal
- Low Engineering: specifies the low engineering value when the raw input is 0 decimal
- Engineering (binary): specifies the V-memory location where the scaled engineering decimal value will be placed

Parameter	DL405 Range
Raw (12 bit binary) . . . . V,P	See DL405 V-memory map - Data Words
High Engineering . . . . K	K0-65535
Low Engineering . . . . K	K0-65535
Engineering (binary) . . . . V,P	See DL405 V-memory map - Data Words

ANSCLB Example

In the following example, the ANSCLB instruction is used to scale a raw value (0-4095 binary) that is in V2000. The engineering scaling range is set 0-1000 (low engineering value - high engineering value). The scaled value will be placed in V2100 in binary format.



```
  Analog Scale 12 Bit Binary to Binary
ANSCLB                           IB-403
  Raw (12 bit binary)             V2000
  High Engineering                K1000
  Low Engineering                    K0
  Engineering (binary)            V2100
```

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

DS5/6	Used
HPP	N/A

Filter Over Time BCD will perform a first-order filter on the Raw Data on a defined time interval. The equation is:

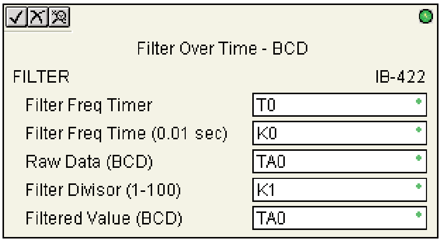
New = Old + [(Raw - Old) / FDC]
where,

New: New Filtered Value

Old: Old Filtered Value

FDC: Filter Divisor Constant

Raw: Raw Data



Filter Over Time - BCD

FILTER IB-422

Filter Freq Timer	T0
Filter Freq Time (0.01 sec)	K0
Raw Data (BCD)	TA0
Filter Divisor (1-100)	K1
Filtered Value (BCD)	TA0

The Filter Divisor Constant is an integer in the range K1 to K100, such that if it equaled K1 then no filtering would be done.

The rate at which the calculation is performed is specified by time in hundredths of a second (0.01 seconds) as the Filter Freq Time parameter. Note that this Timer instruction is embedded in the IBox and must NOT be used anywhere else in your program. Power flow controls whether the calculation is enabled. If it is disabled, the Filter Value is not updated. On the first scan from Program to Run mode, the Filter Value is initialized to 0 to give the calculation a consistent starting point.

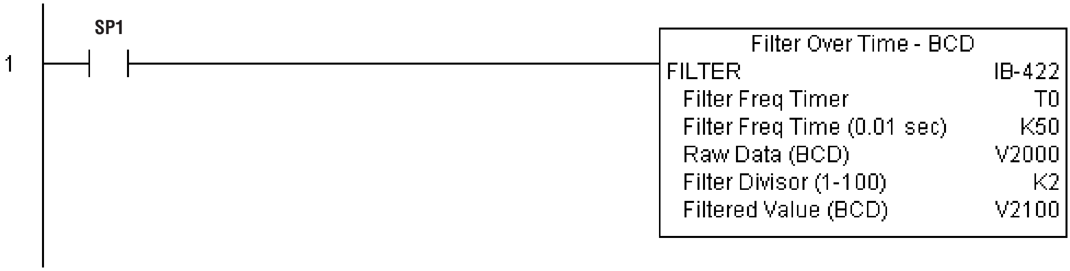
FILTER Parameters

- Filter Frequency Timer: specifies the Timer (T) number which is used by the Filter instruction
- Filter Frequency Time (0.01sec): specifies the rate at which the calculation is performed
- Raw Data (BCD): specifies the V-memory location of the raw unfiltered BCD value
- Filter Divisor (1-100): this constant used to control the filtering effect. A larger value will increase the smoothing effect of the filter. A value of 1 results with no filtering.
- Filtered Value (BCD): specifies the V-memory location where the filtered BCD value will be placed

Parameter	DL405 Range
Filter Frequency Timer . . . . T	T0-377
Filter Frequency Time (0.01 sec) . . . . K	K0-9999
Raw Data (BCD) V	See DL405 V-memory map - Data Words
Filter Divisor (1-100) . . . . K	K1-100
Filtered Value (BCD) V	See DL405 V-memory map - Data Words

FILTER Example

In the following example, the Filter instruction is used to filter a BCD value that is in V2000. Timer(T0) is set to 0.5 sec, the rate at which the filter calculation will be performed. The filter constant is set to 2. A larger value will increase the smoothing effect of the filter. A value of 1 results with no filtering. The filtered value will be placed in V2100.



Filter Over Time - BCD

FILTER IB-422

Filter Freq Timer T0

Filter Freq Time (0.01 sec) K50

Raw Data (BCD) V2000

Filter Divisor (1-100) K2

Filtered Value (BCD) V2100

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

DS5/6	Used
HPP	N/A

Filter Over Time in Binary (decimal) will perform a first-order filter on the Raw Data on a defined time interval. The equation is

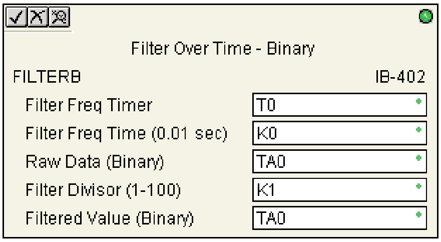
New = Old + [(Raw - Old) / FDC] where

New: New Filtered Value

Old: Old Filtered Value

FDC: Filter Divisor Constant

Raw: Raw Data



Filter Over Time - Binary

FILTERB IB-402

Filter Freq Timer	T0
Filter Freq Time (0.01 sec)	K0
Raw Data (Binary)	TA0
Filter Divisor (1-100)	K1
Filtered Value (Binary)	TA0

The Filter Divisor Constant is an integer in the range K1 to K100, such that if it equaled K1 then no filtering would be done.

The rate at which the calculation is performed is specified by time in hundredths of a second (0.01 seconds) as the Filter Freq Time parameter. Note that this Timer instruction is embedded in the IBox and must NOT be used anywhere else in your program. Power flow controls whether the calculation is enabled. If it is disabled, the Filter Value is not updated. On the first scan from Program to Run mode, the Filter Value is initialized to 0 to give the calculation a consistent starting point.

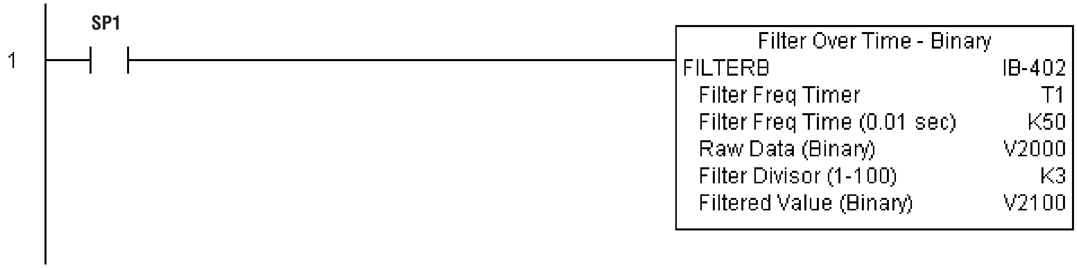
FILTERB Parameters

- Filter Frequency Timer: specifies the Timer (T) number which is used by the Filter instruction
- Filter Frequency Time (0.01sec): specifies the rate at which the calculation is performed
- Raw Data (Binary): specifies the V-memory location of the raw unfiltered binary (decimal) value
- Filter Divisor (1-100): this constant used to control the filtering effect. A larger value will increase the smoothing effect of the filter. A value of 1 results with no filtering.
- Filtered Value (Binary): specifies the V-memory location where the filtered binary (decimal) value will be placed

Parameter	DL405 Range
Filter Frequency Timer . . . . T	T0-377
Filter Frequency Time (0.01 sec) . . . . K	K0-9999
Raw Data (Binary) V	See DL405 V-memory map - Data Words
Filter Divisor (1-100) . . . . K	K1-100
Filtered Value (Binary) V	See DL405 V-memory map - Data Words

FILTERB Example

In the following example, the FILTERB instruction is used to filter a binary value that is in V2000. Timer(T1) is set to 0.5 sec, the rate at which the filter calculation will be performed. The filter constant is set to 3. A larger value will increase the smoothing effect of the filter. A value of 1 results with no filtering. The filtered value will be placed in V2100

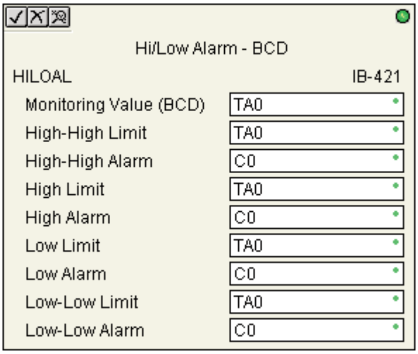


```
Filter Over Time - Binary
FILTERB                          IB-402
  Filter Freq Timer                  T1
  Filter Freq Time (0.01 sec)       K50
  Raw Data (Binary)               V2000
  Filter Divisor (1-100)             K3
  Filtered Value (Binary)         V2100
```

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

DS5/6	Used
HPP	N/A

Hi/Low Alarm - BCD monitors a BCD value V memory location and sets four possible alarm states, High-High, High, Low, and Low-Low whenever the IBox has power flow. You enter the alarm thresholds as constant K BCD values (K0-K9999) and/or BCD value V memory locations.



Hi/Low Alarm - BCD

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

You must ensure that threshold limits are valid, that is HH >= H > L >= LL. Note that when the High-High or Low-Low alarm condition is true, that the High and Low alarms will also be set, respectively. This means you may use the same threshold limit and same alarm bit for the High-High and the High alarms in case you only need one "High" alarm. Also note that the boundary conditions are inclusive. That is, if the Low boundary is K50, and the Low-Low boundary is K10, and if the Monitoring Value equals 10, then the Low Alarm AND the Low-Low alarm will both be ON. If there is no power flow to the IBox, then all alarm bits will be turned off regardless of the value of the Monitoring Value parameter.

HILOAL Parameters

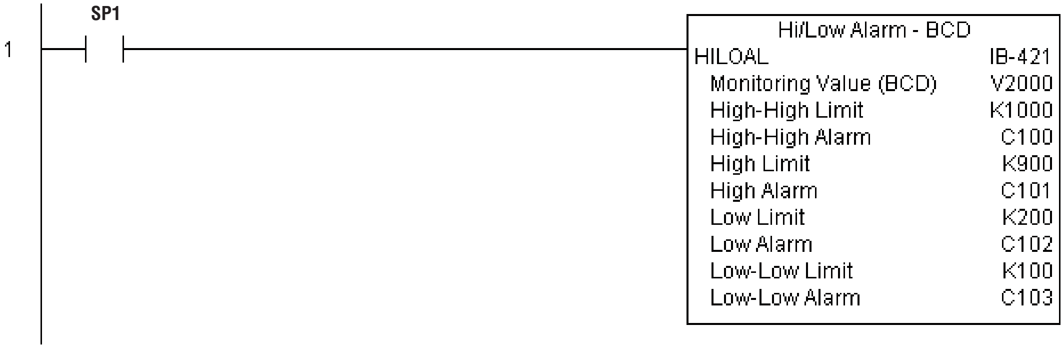
- Monitoring Value (BCD): specifies the V-memory location of the BCD value to be monitored
- High-High Limit: V-memory location or constant specifies the high-high alarm limit
- High-High Alarm: On when the high-high limit is reached
- High Limit: V-memory location or constant specifies the high alarm limit
- High Alarm: On when the high limit is reached
- Low Limit: V-memory location or constant specifies the low alarm limit
- Low Alarm: On when the low limit is reached
- Low-Low Limit: V-memory location or constant specifies the low-low alarm limit
- Low-Low Alarm: On when the low-low limit is reached

Parameter	DL405 Range
Monitoring Value (BCD) V	See DL405 V-memory map - Data Words
High-High Limit . . . . V, K	K0-9999; or see DL405 V-memory map - Data Words
High-High Alarm . . . . X, Y, C, GX,GY, B	See DL405 V-memory map
High Limit . . . . V, K	K0-9999; or see DL405 V-memory map - Data Words
High Alarm. . . . . X, Y, C, GX,GY, B	See DL405 V-memory map
Low Limit . . . . V, K	K0-9999; or see DL405 V-memory map - Data Words
Low Alarm . . . . X, Y, C, GX,GY,B	See DL405 V-memory map
Low-Low Limit . . . . V, K	K0-9999; or see DL405 V-memory map - Data Words
Low-Low Alarm. . . . . X, Y, C, GX,GY, B	See DL405 V-memory map

HILOAL Example

In the following example, the HILOAL instruction is used to monitor a BCD value that is in V2000. If the value in V2000 meets/exceeds the high limit of K900, C101 will turn on. If the value continues to increase to meet/exceed the high-high limit, C100 will turn on. Both bits would be on in this case. The high and high-high limits and alarms can be set to the same value if one "high" limit or alarm is desired to be used.

If the value in V2000 meets or falls below the low limit of K200, C102 will turn on. If the value continues to decrease to meet or fall below the low-low limit of K100, C103 will turn on. Both bits would be on in this case. The low and low-low limits and alarms can be set to the same value if one "low" limit or alarm is desired to be used.



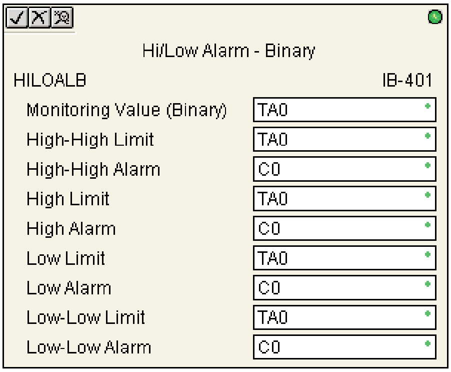
```
           Hi/Low Alarm - BCD
HILOAL                          IB-421
  Monitoring Value (BCD)         V2000
  High-High Limit                K1000
  High-High Alarm                 C100
  High Limit                      K900
  High Alarm                      C101
  Low Limit                       K200
  Low Alarm                       C102
  Low-Low Limit                   K100
  Low-Low Alarm                   C103
```

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

DS5/6	Used
HPP	N/A

Hi/Low Alarm - Binary monitors a binary (decimal) V memory location and sets four possible alarm states, High-High, High, Low, and Low-Low whenever the IBox has power flow. You enter the alarm thresholds as constant K decimal values (K0-K65535) and/or binary (decimal) V memory locations.

You must ensure that threshold limits are valid, that is HH >= H > L >= LL. Note that when the High-High or Low-Low alarm condition is true, that the High and Low alarms will also be set, respectively. This means you may use the same threshold limit and same alarm bit for the High-High and the High alarms in case you only need one "High" alarm. Also note that the boundary conditions are inclusive. That is, if the Low boundary is K50, and the Low-Low boundary is K10, and if the Monitoring Value equals 10, then the Low Alarm AND the Low-Low alarm will both be ON. If there is no power flow to the IBox, then all alarm bits will be turned off regardless of the value of the Monitoring Value parameter.



Hi/Low Alarm - Binary

HILOALB IB-401

Monitoring Value (Binary)	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

HILOALB Parameters

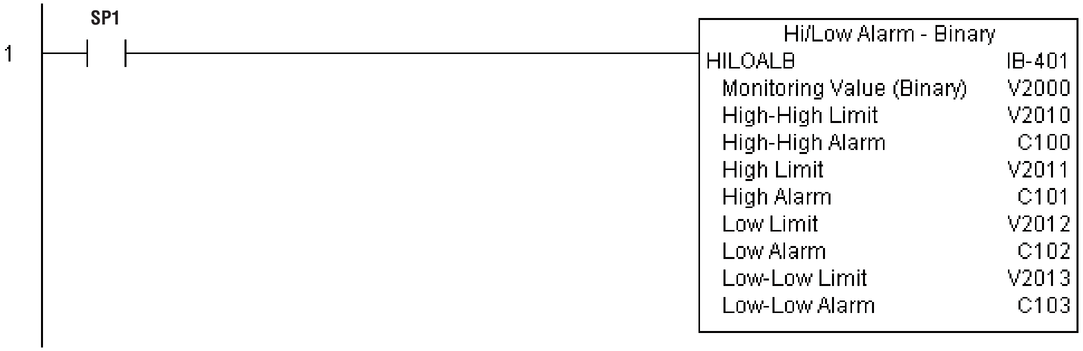
- Monitoring Value (Binary): specifies the V-memory location of the Binary value to be monitored
- High-High Limit: V-memory location or constant specifies the high-high alarm limit
- High-High Alarm: On when the high-high limit is reached
- High Limit: V-memory location or constant specifies the high alarm limit
- High Alarm: On when the high limit is reached
- Low Limit: V-memory location or constant specifies the low alarm limit
- Low Alarm: On when the low limit is reached
- Low-Low Limit: V-memory location or constant specifies the low-low alarm limit
- Low-Low Alarm: On when the low-low limit is reached

Parameter	DL405 Range
Monitoring Value (Binary) V	See DL405 V-memory map - Data Words
High-High Limit . . . . V, K	K0-65535; or see DL405 V-memory map - Data Words
High-High Alarm . . . . X, Y, C, GX,GY, B	See DL405 V-memory map
High Limit . . . . V, K	K0-65535; or see DL405 V-memory map - Data Words
High Alarm. . . . . X, Y, C, GX,GY, B	See DL405 V-memory map
Low Limit . . . . V, K	K0-65535; or see DL405 V-memory map - Data Words
Low Alarm . . . . X, Y, C, GX,GY,B	See DL405 V-memory map
Low-Low Limit . . . . V, K	K0-65535; or see DL405 V-memory map - Data Words
Low-Low Alarm. . . . . X, Y, C, GX,GY, B	See DL405 V-memory map

HILOALB Example

In the following example, the HILOALB instruction is used to monitor a binary value that is in V2000. If the value in V2000 meets/exceeds the high limit of the binary value in V2011, C101 will turn on. If the value continues to increase to meet/exceed the high-high limit value in V2010, C100 will turn on. Both bits would be on in this case. The high and high-high limits and alarms can be set to the same V-memory location/value if one "high" limit or alarm is desired to be used.

If the value in V2000 meets or falls below the low limit of the binary value in V2012, C102 will turn on. If the value continues to decrease to meet or fall below the low-low limit in V2013, C103 will turn on. Both bits would be on in this case. The low and low-low limits and alarms can be set to the same V-memory location/value if one "low" limit or alarm is desired to be used.

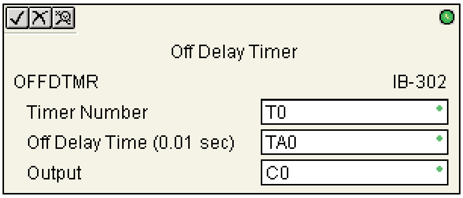


```
Hi/Low Alarm - Binary
HILOALB                         IB-401
  Monitoring Value (Binary)     V2000
  High-High Limit               V2010
  High-High Alarm                C100
  High Limit                    V2011
  High Alarm                     C101
  Low Limit                     V2012
  Low Alarm                      C102
  Low-Low Limit                 V2013
  Low-Low Alarm                  C103
```

Off Delay Timer (OFFDTMR) (IB-302)

DS5/6	Used
HPP	N/A

Off Delay Timer will delay the "turning off" of the Output parameter by the specified Off Delay Time (in hundredths of a second) based on the power flow into the IBox. Once the IBox receives power, the Output bit will turn on immediately. When the power flow to the IBox turns off, the Output bit WILL REMAIN ON for the specified amount of time (in hundredths of a second). Once the Off Delay Time has expired, the output will turn Off. If the power flow to the IBox comes back on BEFORE the Off Delay Time, then the timer is RESET and the Output will remain On - so you must continuously have NO power flow to the IBox for AT LEAST the specified Off Delay Time before the Output will turn Off.



Off Delay Timer

OFFDTMR IB-302

Timer Number: T0

Off Delay Time (0.01 sec): TA0

Output: C0

This IBox utilizes a Timer resource (TMRF), which cannot be used anywhere else in your program.

OFFDTMR Parameters

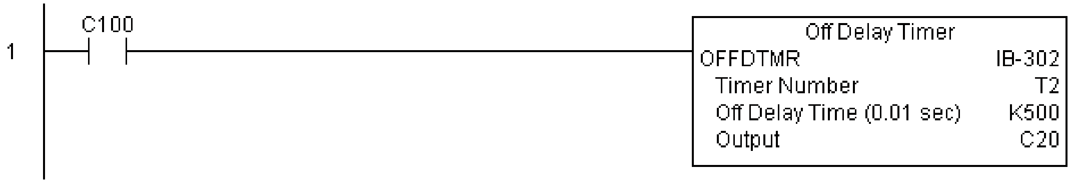
- Timer Number: specifies the Timer(TMRF) number which is used by the OFFDTMR instruction
- Off Delay Time (0.01sec): specifies how long the Output will remain on once power flow to the Ibox is removed
- Output: specifies the output that will be delayed “turning off” by the Off Delay Time.

Parameter	DL405 Range
Timer Number . . . . T	T0-377
Off Delay Time . . . . K,V	K0-9999; See DL405 V-memory map - Data Words
Output . . . . X, Y, C, GX,GY, B	See DL405 V-memory map

OFFDTMR Example

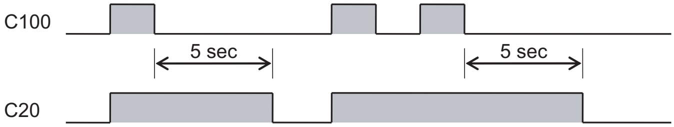
In the following example, the OFFDTMR instruction is used to delay the “turning off”of output C20. Timer 2 (T2) is set to 5 seconds, the “off-delay” period.

When C100 turns on, C20 turns on and will remain on while C100 is on. When C100 turns off, C20 will remain for the specified Off Delay Time (5s), and then turn off.



Off Delay Timer
OFFDTMR IB-302
Timer Number T2
Off Delay Time (0.01 sec) K500
Output C20

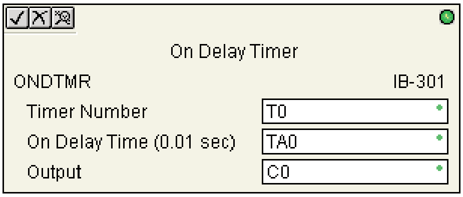
Example timing diagram



On Delay Timer (ONDTMR) (IB-301)

DS5/6	Used
HPP	N/A

On Delay Timer will delay the "turning on" of the Output parameter by the specified amount of time (in hundredths of a second) based on the power flow into the IBox. Once the IBox loses power, the Output is turned off immediately. If the power flow turns off BEFORE the On Delay Time, then the timer is RESET and the Output is never turned on, so you must have continuous power flow to the IBox for at least the specified On Delay Time before the Output turns On.


On Delay Timer

ONDTMR IB-301

Timer Number: T0

On Delay Time (0.01 sec): TA0

Output: C0

This IBox utilizes a Timer resource (TMRF), which cannot be used anywhere else in your program.

ONDTMR Parameters

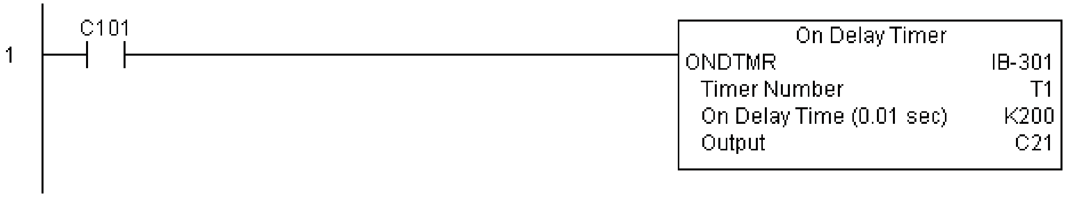
- Timer Number: specifies the Timer(TMRF) number which is used by the ONDTMR instruction
- On Delay Time (0.01sec): specifies how long the Output will remain off once power flow to the Ibox is applied
- Output: specifies the output that will be delayed “turning on” by the On Delay Time

Parameter	DL405 Range
Timer Number . . . . T	T0-377
On Delay Time . . . . K,V	K0-9999; See DL405 V-memory map - Data Words
Output . . . . X, Y, C, GX,GY, B	See DL405 V-memory map

ONDTMR Example

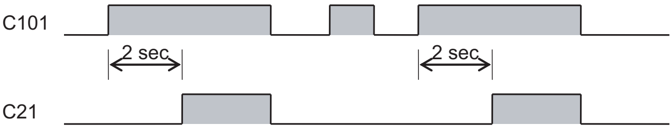
In the following example, the ONDTMR instruction is used to delay the “turning on” of output C21. Timer 1 (T1) is set to 2 seconds, the “on-delay” period.

When C101 turns on, C21 is delayed turning on by 2 seconds. When C101 turns off, C21 turns off imediately.



```
On Delay Timer
ONDTMR                        IB-301
  Timer Number                    T1
  On Delay Time (0.01 sec)      K200
  Output                         C21
```

Example timing diagram



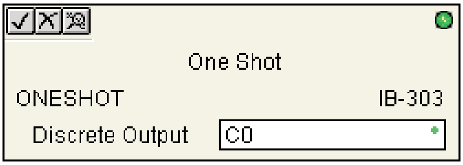
One Shot (ONESHOT) (IB-303)

DS5/6	Used
HPP	N/A

One Shot will turn on the given bit output parameter for one scan on an OFF to ON transition of the power flow into the IBox. This IBox is simply a different name for the PD Coil (Positive Differential).

ONESHOT Parameters

- Discrete Output: specifies the output that will be on for one scan



One Shot

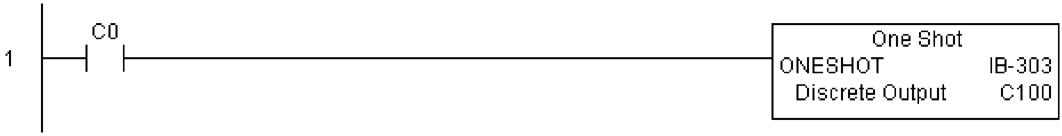
ONESHOT IB-303

Discrete Output: C0

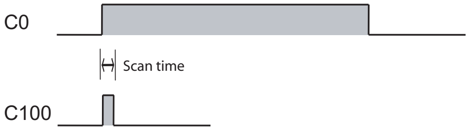
Parameter	DL405 Range
Discrete Output . . . . X, Y, C	See DL405 V-memory map

ONESHOT Example

In the following example, the ONESHOT instruction is used to turn C100 on for one PLC scan after C0 goes from an off to on transition. The input logic must produce an off to on transition to execute the One Shot instruction.



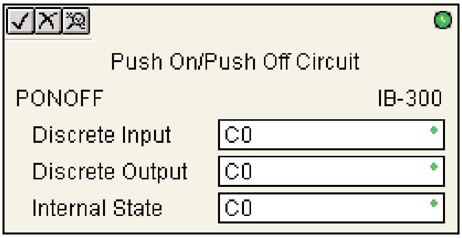
Example timing diagram



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

DS5/6	Used
HPP	N/A

Push On/Push Off Circuit toggles an output state whenever its input power flow transitions from off to on. Requires an extra bit parameter for scan-to-scan state information. This extra bit must NOT be used anywhere else in the program. This is also known as a "flip-flop circuit".


Push On/Push Off Circuit

PONOFF IB-300

Discrete Input: C0

Discrete Output: C0

Internal State: C0

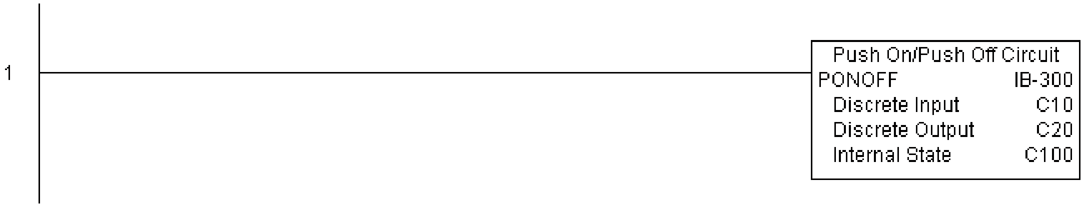
PONOFF Parameters

- Discrete Input: specifies the input that will toggle the specified output
- Discrete Output: specifies the output that will be "turned on/off" or toggled
- Internal State: specifies a work bit that is used by the instruction

Parameter	DL405 Range
Discrete Input . . . . X,Y,C,S,T,CT,GX,GY,SP,B,PB	See DL405 V-memory map
Discrete Output . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Internal State . . . . X, Y, C	See DL405 V-memory map

PONOFF Example

In the following example, the PONOFF instruction is used to control the on and off states of the output C20 with a single input C10. When C10 is pressed once, C20 turns on. When C10 is pressed again, C20 turns off. C100 is an internal bit used by the instruction.



```
Push On/Push Off Circuit
PONOFF               IB-300
  Discrete Input        C10
  Discrete Output       C20
  Internal State       C100
```

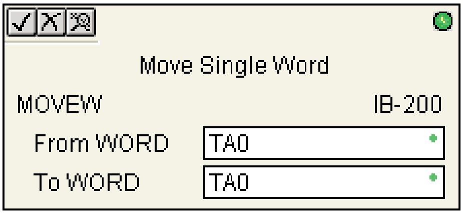
Move Single Word (MOVEW) (IB-200)

DS5/6	Used
HPP	N/A

Move Single Word moves (copies) a word to a memory location directly or indirectly via a pointer, either as a HEX constant, from a memory location, or indirectly through a pointer.

MOVEW Parameters

- From WORD: specifies the word that will be moved to another location
- To WORD: specifies the location where the “From WORD” will be moved to



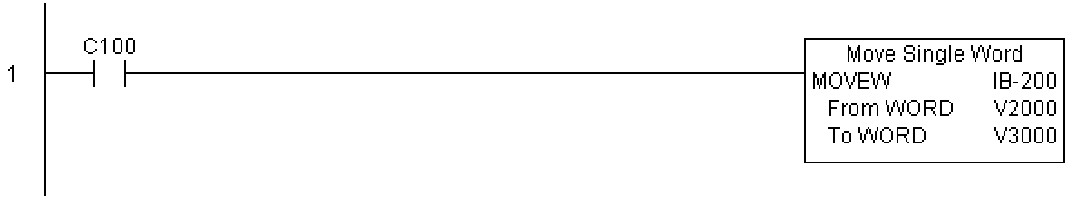
From WORD: TA0

To WORD: TA0

Parameter	DL405 Range
From WORD . . . . V,P,K	K0-FFFF; See DL405 V-memory map - Data Words
To WORD. . . . . V,P	See DL405 V-memory map - Data Words

MOVEW Example

In the following example, the MOVEW instruction is used to move 16-bits of data from V2000 to V3000 when C100 turns on.

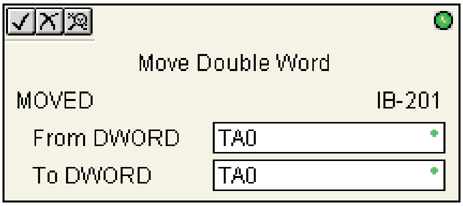


Move Single Word
MOVEW IB-200
From WORD V2000
To WORD V3000

Move Double Word (MOVED) (IB-201)

DS5/6	Used
HPP	N/A

Move Double Word moves (copies) a double word to two consecutive memory locations directly or indirectly via a pointer, either as a double HEX constant, from a double memory location, or indirectly through a pointer to a double memory location.



Move Double Word

MOVED IB-201

From DWORD TA0

To DWORD TA0

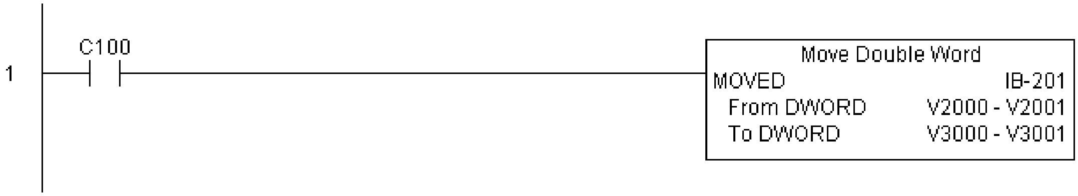
MOVED Parameters

- From DWORD: specifies the double word that will be moved to another location
- To DWORD: specifies the location where the “From DWORD” will be moved to

Parameter	DL405 Range
From DWORD .... V,P,K	K0-FFFFFFFF; See DL405 V-memory map - Data Words
To DWORD .... V,P	See DL405 V-memory map - Data Words

MOVED Example

In the following example, the MOVED instruction is used to move 32-bits of data from V2000 and V2001 to V3000 and V3001 when C100 turns on.

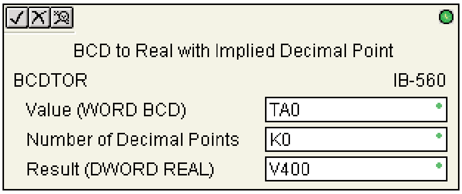


BCD to Real with Implied Decimal Point (BCDTOR) (IB-560)

DS5/6	Used
HPP	N/A

BCD to Real with Implied Decimal Point converts the given 4 digit WORD BCD value to a Real number, with the implied number of decimal points (K0-K4).

For example, BCDTOR K1234 with an implied number of decimal points equal to K1, would yield R123.4


BCD to Real with Implied Decimal Point
BCDTOR IB-560
Value (WORD BCD) TA0
Number of Decimal Points K0
Result (DWORD REAL) V400

BCDTOR Parameters

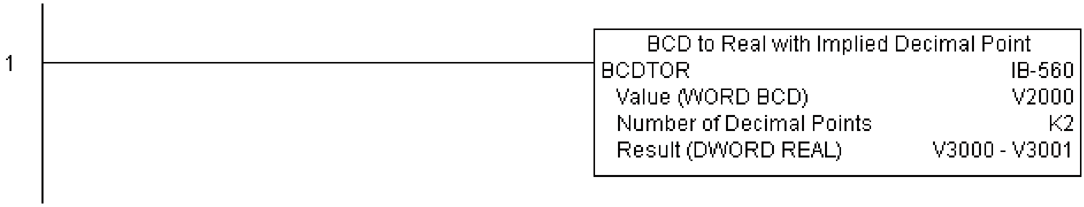
- Value (WORD BCD): specifies the word or constant that will be converted to a Real number
- Number of Decimal Points: specifies the number of implied decimal points in the Result DWORD
- Result (DWORD REAL): specifies the location where the Real number will be placed

Parameter	DL405 Range
Value (WORD BCD) . . . . V,P,K	K0-9999; See DL405 V-memory map - Data Words
Number of Decimal Points . . . . K	K0-4
Result (DWORD REAL) V	See DL405 V-memory map - Data Words

BCDTOR Example

In the following example, the BCDTOR instruction is used to convert the 16-bit data in V2000 from a 4-digit BCD data format to a 32-bit REAL (floating point) data format and stored into V3000 and V3001.

K2 in the Number of Decimal Points implies the data will have two digits to the right of the decimal point.



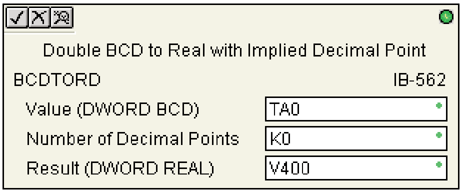
```
1   BCD to Real with Implied Decimal Point
    BCDTOR                                IB-560
      Value (WORD BCD)                     V2000
      Number of Decimal Points                K2
      Result (DWORD REAL)          V3000 - V3001
```

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

DS5/6	Used
HPP	N/A

Double BCD to Real with Implied Decimal Point converts the given 8 digit DWORD BCD value to a Real number, given an implied number of decimal points (K0-K8).

For example, BCDTORD K12345678 with an implied number of decimal points equal to K5, would yield R123.45678


Double BCD to Real with Implied Decimal Point

BCDTORD IB-562

Value (DWORD BCD): TA0

Number of Decimal Points: K0

Result (DWORD REAL): V400

BCDTORD Parameters

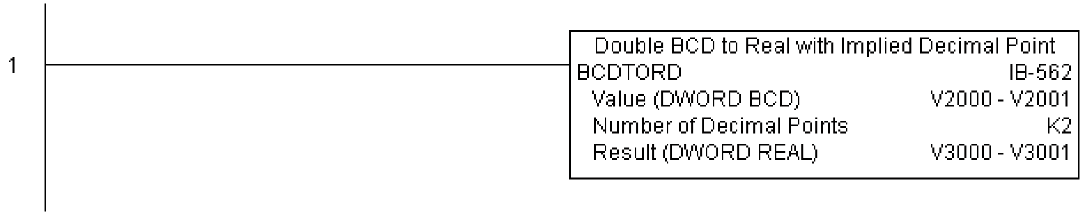
- Value (DWORD BCD): specifies the Dword or constant that will be converted to a Real number
- Number of Decimal Points: specifies the number of implied decimal points in the Result DWORD
- Result (DWORD REAL): specifies the location where the Real number will be placed

Parameter	DL405 Range
Value (DWORD BCD) . . . . V,P,K	K0-99999999; See DL405 V-memory map - Data Words
Number of Decimal Points . . . . K	K0-8
Result (DWORD REAL) V	See DL405 V-memory map - Data Words

BCDTORD Example

In the following example, the BCDTORD instruction is used to convert the 32-bit data in V2000 from an 8-digit BCD data format to a 32-bit REAL (floating point) data format and stored into V3000 and V3001.

K2 in the Number of Decimal Points implies the data will have two digits to the right of the decimal point.

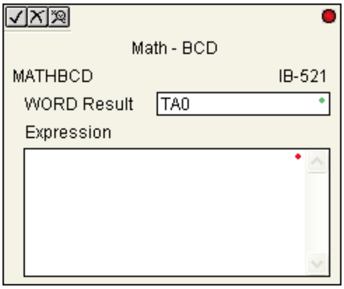


```
 Double BCD to Real with Implied Decimal Point
BCDTORD                                  IB-562
  Value (DWORD BCD)               V2000 - V2001
  Number of Decimal Points                   K2
  Result (DWORD REAL)             V3000 - V3001
```

Math - BCD (MATHBCD) (IB-521)

DS5/6	Used
HPP	N/A

Math - BCD Format lets you enter complex mathematical expressions like you would in Visual Basic, Excel, or C++ to do complex calculations, nesting parentheses up to 4 levels deep. In addition to + - * /, you can do Modulo (% aka Remainder), Bit-wise And (&) Or (|) Xor (^), and some BCD functions - Convert to BCD (BCD), Convert to Binary (BIN), BCD Complement (BCDCPL), Convert from Gray Code (GRAY), Invert Bits (INV), and BCD/HEX to Seven Segment Display (SEG).



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

Every V-memory reference MUST be to a single word BCD formatted value. Intermediate results can go up to 32 bit values, but as long as the final result fits in a 16 bit BCD word, the calculation is valid. Typical example of this is scaling using multiply then divide, (V2000 * K1000) / K4095. The multiply term most likely will exceed 9999 but fits within 32 bits. The divide operation will divide 4095 into the 32-bit accumulator, yielding a result that will always fit in 16 bits.

You can reference binary V-memory values by using the BCD conversion function on a V memory location but NOT an expression. That is, BCD(V2000) is okay and will convert V2000 from Binary to BCD, but BCD(V2000 + V3000) will add V2000 as BCD, to V3000 as BCD, then interpret the result as Binary and convert it to BCD - NOT GOOD.

Also, the final result is a 16 bit BCD number and so you could do BIN around the entire operation to store the result as Binary.

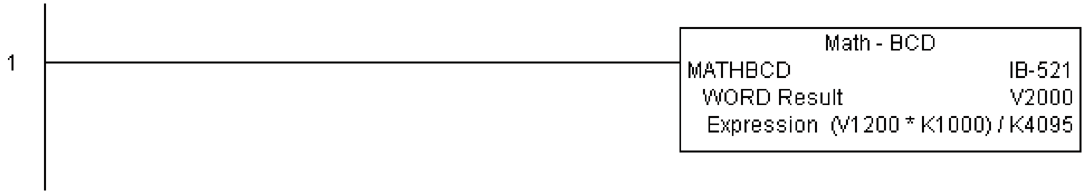
MATHBCD Parameters

- WORD Result: specifies the location where the BCD result of the mathematical expression will be placed (result must fit into 16 bit single V-memory location)
- Expression: specifies the mathematical expression to be executed and the result is stored in specified WORD Result. Each V-memory location used in the expression must be in BCD format.

Parameter	DL405 Range
WORD Result V	See DL405 V-memory map - Data Words
Expression	Text

MATHBCD Example

In the following example, the MATHBCD instruction is used to calculate the math expression which multiplies the BCD value in V1200 by 1000 then divides by 4095 and loads the resulting value in V2000.

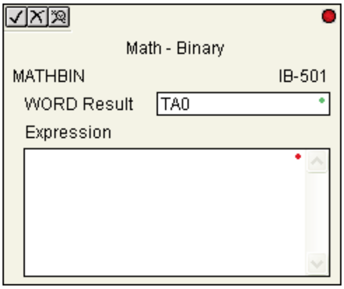


```
1
                         Math - BCD
                 MATHBCD                  IB-521
                   WORD Result             V2000
                   Expression  (V1200 * K1000) / K4095
```

Math - Binary (MATHBIN) (IB-501)

DS5/6	Used
HPP	N/A

Math - Binary Format lets you enter complex mathematical expressions like you would in Visual Basic, Excel, or C++ to do complex calculations, nesting parentheses up to 4 levels deep. In addition to + - * /, you can do Modulo (% aka Remainder), Shift Right (>>) and Shift Left (<<), Bit-wise And (&) Or (|) Xor (^), and some binary functions - Convert to BCD (BCD), Convert to Binary (BIN), Decode Bits (DECO), Encode Bits (ENCO), Invert Bits (INV), HEX to Seven Segment Display (SEG), and Sum Bits (SUM).



Math - Binary

MATHBIN IB-501

WORD Result TA0

Expression

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

Every V-memory reference MUST be to a single word binary formatted value. Intermediate results can go up to 32 bit values, but as long as the final result fits in a 16 bit binary word, the calculation is valid. Typical example of this is scaling using multiply then divide, (V2000 * K1000) / K4095. The multiply term most likely will exceed 65535 but fits within 32 bits. The divide operation will divide 4095 into the 32-bit accumulator, yielding a result that will always fit in 16 bits.

You can reference BCD V memory values by using the BIN conversion function on a V-memory location but NOT an expression. That is, BIN(V2000) is okay and will convert V2000 from BCD to Binary, but BIN(V2000 + V3000) will add V2000 as Binary, to V3000 as Binary, then interpret the result as BCD and convert it to Binary - NOT GOOD.

Also, the final result is a 16 bit binary number and so you could do BCD around the entire operation to store the result as BCD.

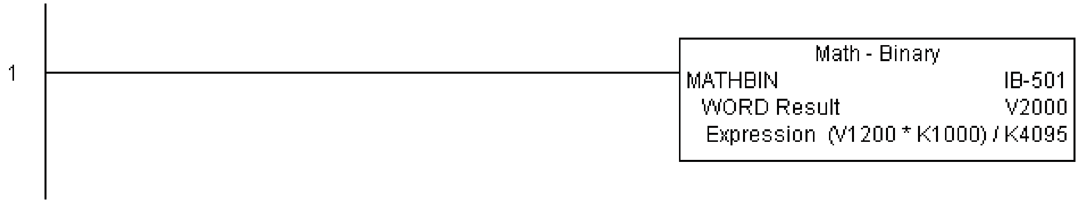
MATHBIN Parameters

- WORD Result: specifies the location where the binary result of the mathematical expression will be placed (result must fit into 16 bit single V-memory location)
- Expression: specifies the mathematical expression to be executed and the result is stored in specified WORD Result. Each V-memory location used in the expression must be in binary format.

Parameter	DL405 Range
WORD Result V	See DL405 V-memory map - Data Words
Expression	Text

MATHBIN Example

In the following example, the MATHBIN instruction is used to calculate the math expression which multiplies the Binary value in V1200 by 1000 then divides by 4095 and loads the resulting value in V2000.

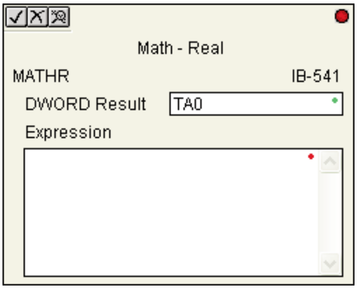


Math - Binary
MATHBIN IB-501
WORD Result V2000
Expression (V1200 * K1000) / K4095

Math - Real (MATHR) (IB-541)

DS5/6	Used
HPP	N/A

Math - Real Format lets you enter complex mathematical expressions like you would in Visual Basic, Excel, or C++ to do complex calculations, nesting parentheses up to 4 levels deep. In addition to + - * /, you can do Bit-wise And (&) Or (|) Xor (^), and many Real functions - Arc Cosine (ACOSR), Arc Sine (ASINR), Arc Tangent (ATANR), Cosine (COSR), Convert Radians to Degrees (DEGR), Invert Bits (INV), Convert Degrees to Radians (RADR), HEX to Seven Segment Display (SEG), Sine (SINR), Square Root (SQRTR), Tangent (TANR).


Math - Real
MATHR IB-541
DWORD Result: TA0
Expression

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

Every V-memory reference MUST be able to fit into a double word Real formatted value.

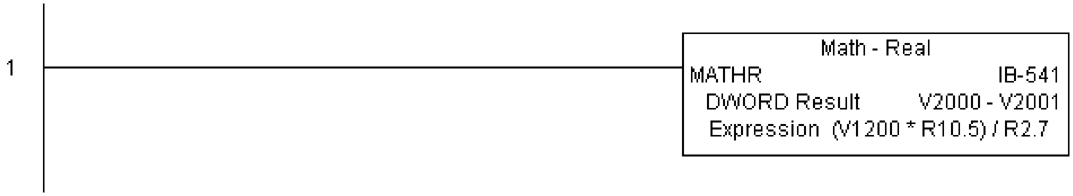
MATHR Parameters

- DWORD Result: specifies the location where the Real result of the mathematical expression will be placed (result must fit into a double word Real formatted location)
- Expression: specifies the mathematical expression to be executed and the result is stored in specified DWORD Result location. Each V-memory location used in the expression must be in Real format.

Parameter	DL405 Range
DWORD Result V	See DL405 V-memory map - Data Words
Expression	Text

MATHR Example

In the following example, the MATHR instruction is used to calculate the math expression which multiplies the REAL (floating point) value in V1200 by 10.5 then divides by 2.7 and loads the resulting 32-bit value in V2000 and V2001.



```
1   Math - Real
    MATHR                            IB-541
      DWORD Result          V2000 - V2001
      Expression  (V1200 * R10.5) / R2.7
```

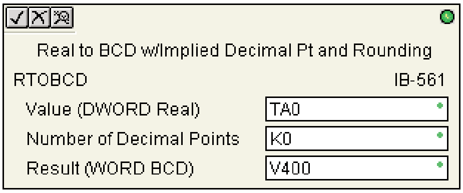
Real to BCD with Implied Decimal Point and Rounding (RTOBCD) (IB-561)

DS5/6	Used
HPP	N/A

Real to BCD with Implied Decimal Point and Rounding converts the absolute value of the given Real number to a 4 digit BCD number, compensating for an implied number of decimal points (K0-K4) and performs rounding.

For example, RTOBCD R56.74 with an implied number of decimal points equal to K1, would yield 567 BCD. If the implied number of decimal points was 0, then the function would yield 57 BCD (note that it rounded up).

If the Real number is negative, the Result will equal its positive, absolute value.



Real to BCD w/Implied Decimal Pt and Rounding

RTOBCD IB-561

Value (DWORD Real) TA0

Number of Decimal Points K0

Result (WORD BCD) V400

RTOBCD Parameters

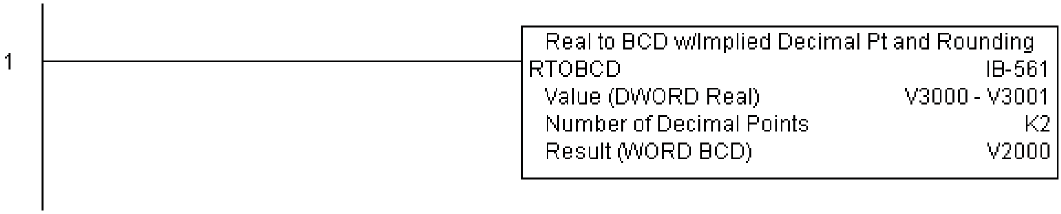
- Value (DWORD Real): specifies the Real Dword location or number that will be converted and rounded to a BCD number with decimal points
- Number of Decimal Points: specifies the number of implied decimal points in the Result WORD
- Result (WORD BCD): specifies the location where the rounded/implied decimal points BCD value will be placed

Parameter	DL405 Range
Value (DWORD Real) . . . . V,P,R	R ; See DL405 V-memory map - Data Words
Number of Decimal Points . . . . K	K0-4
Result (WORD BCD) V	See DL405 V-memory map - Data Words

RTOBCD Example

In the following example, the RTOBCD instruction is used to convert the 32-bit REAL (floating point) data format in V3000 and V3001 to the 4-digit BCD data format and stored in V2000.

K2 in the Number of Decimal Points implies the data will have two implied decimal points.



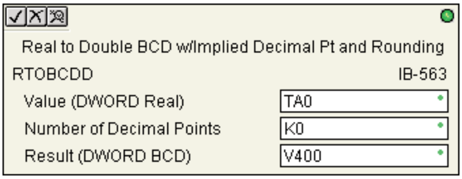
```
1   Real to BCD w/Implied Decimal Pt and Rounding
    RTOBCD                                        IB-561
      Value (DWORD Real)                   V3000 - V3001
      Number of Decimal Points                        K2
      Result (WORD BCD)                            V2000
```

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

DS5/6	Used
HPP	N/A

Real to Double BCD with Implied Decimal Point and Rounding converts the absolute value of the given Real number to an 8 digit DWORD BCD number, compensating for an implied number of decimal points (K0-K8) and performs rounding.

For example, RTOBCDD R38156.74 with an implied number of decimal points equal to K1, would yield 381567 BCD. If the implied number of decimal points was 0, then the function would yield 38157 BCD (note that it rounded up).


Real to Double BCD w/Implied Decimal Pt and Rounding
RTOBCDD IB-563
Value (DWORD Real) TA0
Number of Decimal Points K0
Result (DWORD BCD) V400

If the Real number is negative, the Result will equal its positive, absolute value.

RTOBCDD Parameters

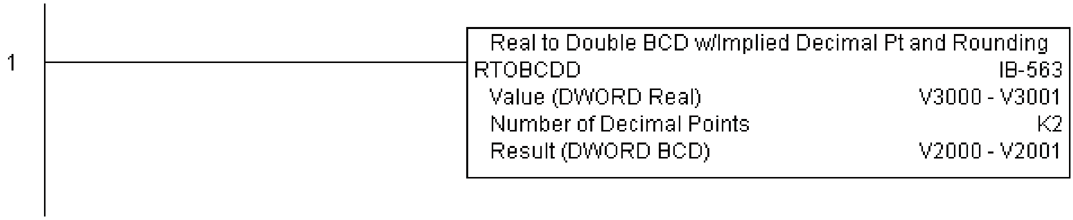
- Value (DWORD Real): specifies the Dword Real number that will be converted and rounded to a BCD number with decimal points
- Number of Decimal Points: specifies the number of implied decimal points in the Result DWORD
- Result (DWORD BCD): specifies the location where the rounded/implied decimal points DWORD BCD value will be placed

Parameter	DL405 Range
Value (DWORD Real) . . . . V,P,R	R ; See DL405 V-memory map - Data Words
Number of Decimal Points . . . . K	K0-8
Result (DWORD BCD) V	See DL405 V-memory map - Data Words

RTOBCDD Example

In the following example, the RTOBCDD instruction is used to convert the 32-bit REAL (floating point) data format in V3000 and V3001 to the 8-digit BCD data format and stored in V2000 and V2001.

K2 in the Number of Decimal Points implies the data will have two implied decimal points.



Real to Double BCD w/Implied Decimal Pt and Rounding
RTOBCDD IB-563
Value (DWORD Real) V3000 - V3001
Number of Decimal Points K2
Result (DWORD BCD) V2000 - V2001

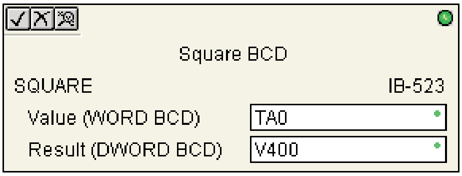
Square BCD (SQUARE) (IB-523)

DS5/6	Used
HPP	N/A

Square BCD squares the given 4-digit WORD BCD number and writes it in as an 8-digit DWORD BCD result.

SQUARE Parameters

- Value (WORD BCD): specifies the BCD Word or constant that will be squared
- Result (DWORD BCD): specifies the location where the squared DWORD BCD value will be placed



Square BCD

SQUARE IB-523

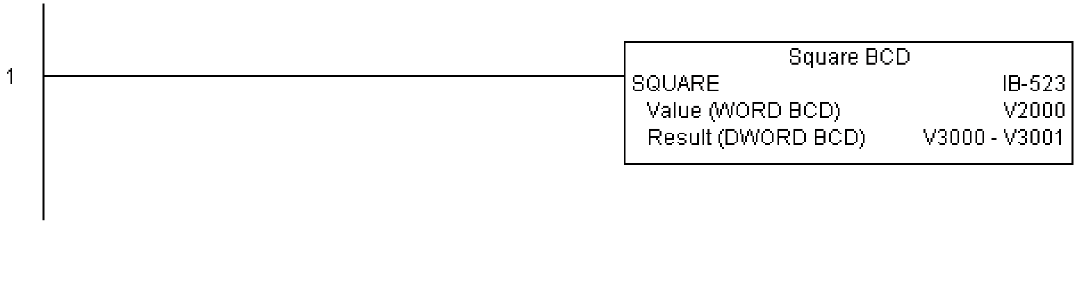
Value (WORD BCD): TA0

Result (DWORD BCD): V400

Parameter	DL405 Range
Value (WORD BCD) . . . . V,P,K	K0-9999 ; See DL405 V-memory map - Data Words
Result (DWORD BCD) V	See DL405 V-memory map - Data Words

SQUARE Example

In the following example, the SQUARE instruction is used to square the 4-digit BCD value in V2000 and store the 8-digit double word BCD result in V3000 and V3001



```
1   Square BCD
    SQUARE                        IB-523
      Value (WORD BCD)             V2000
      Result (DWORD BCD)   V3000 - V3001
```

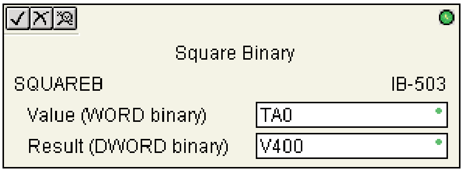
Square Binary (SQUAREB) (IB-503)

DS5/6	Used
HPP	N/A

Square Binary squares the given 16-bit WORD Binary number and writes it as a 32-bit DWORD Binary result.

SQUAREB Parameters

- Value (WORD Binary): specifies the binary Word or constant that will be squared
- Result (DWORD Binary): specifies the location where the squared DWORD binary value will be placed



Square Binary

SQUAREB IB-503

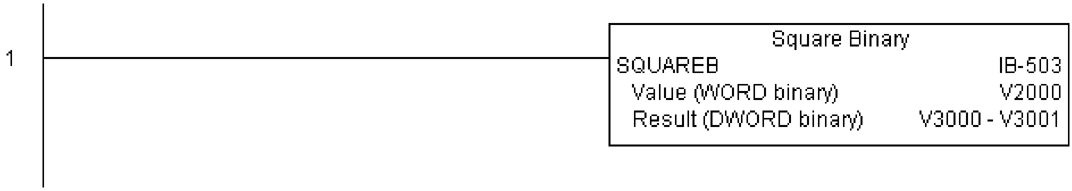
Value (WORD binary) TA0

Result (DWORD binary) V400

Parameter	DL405 Range
Value (WORD Binary) . . . . V,P,K	K0-65535; See DL405 V-memory map - Data Words
Result (DWORD Binary) V	See DL405 V-memory map - Data Words

SQUAREB Example

In the following example, the SQUAREB instruction is used to square the single word Binary value in V2000 and store the 8-digit double word Binary result in V3000 and V3001.



```
1
                         Square Binary
SQUAREB                                     IB-503
  Value (WORD binary)                        V2000
  Result (DWORD binary)              V3000 - V3001
```

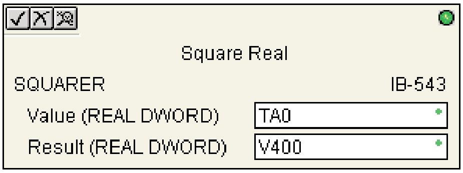
Square Real (SQUARER) (IB-543)

DS5/6	Used
HPP	N/A

Square Real squares the given REAL DWORD number and writes it to a REAL DWORD result.

SQUARER Parameters

- Value (REAL DWORD): specifies the Real DWORD location or number that will be squared
- Result (REAL DWORD): specifies the location where the squared Real DWORD value will be placed



Square Real

SQUARER IB-543

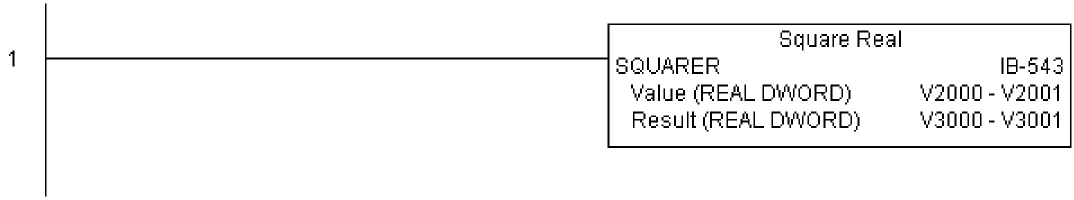
Value (REAL DWORD): TA0

Result (REAL DWORD): V400

Parameter	DL405 Range
Value (REAL DWORD) . . . . V,P,R	R ; See DL405 V-memory map - Data Words
Result (REAL DWORD) V	See DL405 V-memory map - Data Words

SQUARER Example

In the following example, the SQUARER instruction is used to square the 32-bit floating point REAL value in V2000 and V2001 and store the REAL value result in V3000 and V3001.



```
1
                              Square Real
                  SQUARER                            IB-543
                    Value (REAL DWORD)          V2000 - V2001
                    Result (REAL DWORD)         V3000 - V3001
```

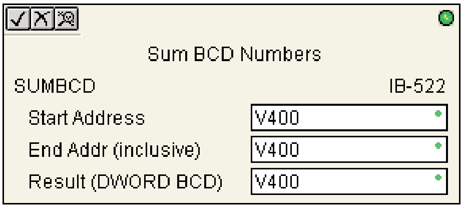
Sum BCD Numbers (SUMBCD) (IB-522)

DS5/6	Used
HPP	N/A

Sum BCD Numbers sums up a list of consecutive 4-digit WORD BCD numbers into an 8-digit DWORD BCD result.

You specify the group's starting and ending V- memory addresses (inclusive). When enabled, this instruction will add up all the numbers in the group (so you may want to place a differential contact driving the enable).

SUMBCD could be used as the first part of calculating an average.



Sum BCD Numbers

SUMBCD IB-522

Start Address: V400

End Addr (inclusive): V400

Result (DWORD BCD): V400

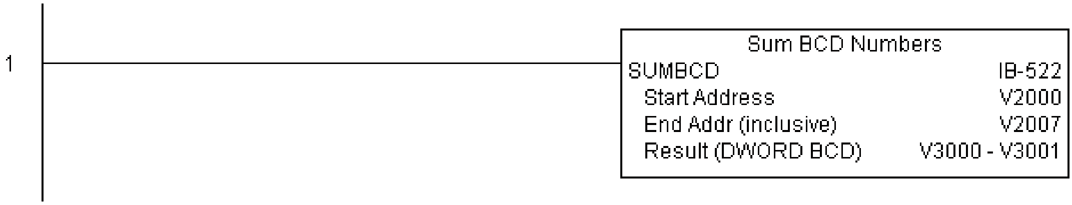
SUMBCD Parameters

- Start Address: specifies the starting address of a block of V-memory location values to be added together (BCD)
- End Addr (inclusive): specifies the ending address of a block of V-memory location values to be added together (BCD)
- Result (DWORD BCD): specifies the location where the sum of the block of V-memory BCD values will be placed

Parameter	DL405 Range
Start Address V	See DL405 V-memory map - Data Words
End Address (inclusive) V	See DL405 V-memory map - Data Words
Result (DWORD BCD) V	See DL405 V-memory map - Data Words

SUMBCD Example

In the following example, the SUMBCD instruction is used to total the sum of all BCD values in words V2000 thru V2007 and store the resulting 8-digit double word BCD value in V3000 and V3001.



```
1
                              Sum BCD Numbers
                      SUMBCD                        IB-522
                        Start Address               V2000
                        End Addr (inclusive)        V2007
                        Result (DWORD BCD)   V3000 - V3001
```

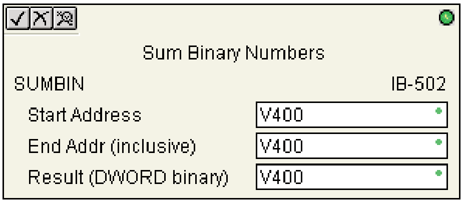
Sum Binary Numbers (SUMBIN) (IB-502)

DS5/6	Used
HPP	N/A

Sum Binary Numbers sums up a list of consecutive 16-bit WORD Binary numbers into a 32-bit DWORD binary result.

You specify the group's starting and ending V- memory addresses (inclusive). When enabled, this instruction will add up all the numbers in the group (so you may want to place a differential contact driving the enable).

SUMBIN could be used as the first part of calculating an average.


Sum Binary Numbers
SUMBIN IB-502
Start Address V400
End Addr (inclusive) V400
Result (DWORD binary) V400

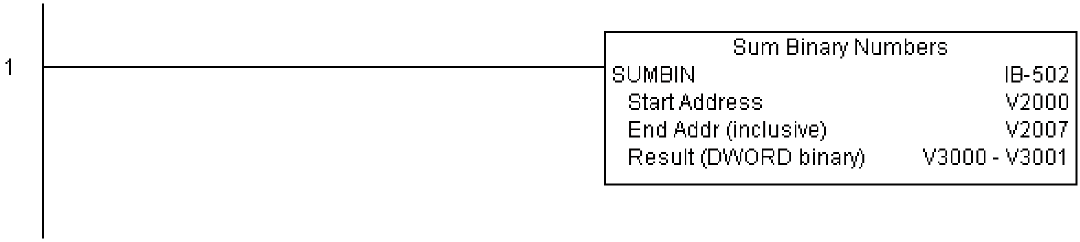
SUMBIN Parameters

- Start Address: specifies the starting address of a block of V-memory location values to be added together (Binary)
- End Addr (inclusive): specifies the ending address of a block of V-memory location values to be added together (Binary)
- Result (DWORD Binary): specifies the location where the sum of the block of V-memory binary values will be placed

Parameter	DL405 Range
Start Address V	See DL405 V-memory map - Data Words
End Address (inclusive) V	See DL405 V-memory map - Data Words
Result (DWORD Binary) V	See DL405 V-memory map - Data Words

SUMBIN Example

In the following example, the SUMBIN instruction is used to total the sum of all Binary values in words V2000 thru V2007 and store the resulting 8-digit double word Binary value in V3000 and V3001.

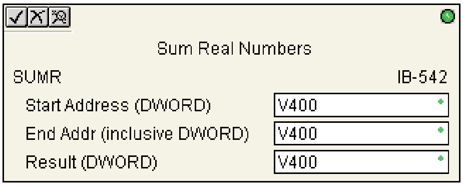


Sum Real Numbers (SUMR) (IB-542)

DS5/6	Used
HPP	N/A

Sum Real Numbers sums up a list of consecutive REAL DWORD numbers into a REAL DWORD result.

You specify the group's starting and ending V- memory addresses (inclusive).


Sum Real Numbers
SUMR IB-542
Start Address (DWORD) V400
End Addr (inclusive DWORD) V400
Result (DWORD) V400

Remember that Real numbers are DWORDs and occupy 2 words of V memory each, so the number of Real values summed up is equal to half the number of memory locations. Note that the End Address can be EITHER word of the 2 word ending address, for example, if you wanted to add the 4 Real numbers stored in V2000 thru V2007 (V2000, V2002, V2004, and V2006), you can specify V2006 OR V2007 for the ending address and you will get the same result.

When enabled, this instruction will add up all the numbers in the group (so you may want to place a differential contact driving the enable).

SUMR could be used as the first part of calculating an average.

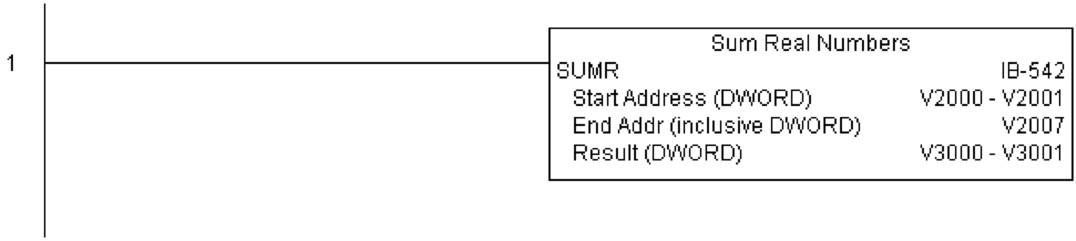
SUMR Parameters

- Start Address (DWORD): specifies the starting address of a block of V-memory location values to be added together (Real)
- End Addr (inclusive) (DWORD): specifies the ending address of a block of V-memory location values to be added together (Real)
- Result (DWORD): specifies the location where the sum of the block of V-memory Real values will be placed

Parameter	DL405 Range
Start Address (inclusive DWORD) V	See DL405 V-memory map - Data Words
End Address (inclusive DWORD) V	See DL405 V-memory map - Data Words
Result (DWORD) V	See DL405 V-memory map - Data Words

SUMR Example

In the following example, the SUMR instruction is used to total the sum of all floating point REAL number values in words V2000 thru V2007 and store the resulting 32-bit floating point REAL number value in V3000 and V3001.

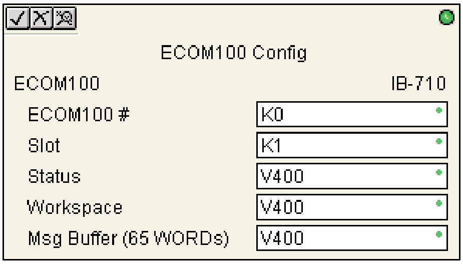


```
Sum Real Numbers
SUMR                                   IB-542
  Start Address (DWORD)           V2000 - V2001
  End Addr (inclusive DWORD)              V2007
  Result (DWORD)                  V3000 - V3001
```

ECOM100 Configuration (ECOM100) (IB-710)

DS5/6	Used
HPP	N/A

ECOM100 Configuration defines all the common information for one specific ECOM100 module which is used by the other ECOM100 IBoxes; for example, ECRX - ECOM100 Network Read , ECEMAIL - ECOM100 Send EMail, ECIPSUP - ECOM100 IP Setup, etc.



ECOM100 Config

ECOM100 IB-710

ECOM100 #	K0
Slot	K1
Status	V400
Workspace	V400
Msg Buffer (65 WORDs)	V400

You MUST have the ECOM100 Configuration IBox at the top of your ladder/stage program with any other configuration IBoxes. The Message Buffer parameter specifies the starting address of a 65 WORD buffer. This is 101 Octal addresses (e.g. V1400 thru V1500).

If you have more than one ECOM100 in your PLC, you must have a different ECOM100 Configuration IBox for EACH ECOM100 module in your system that utilizes any ECOM IBox instructions.

The Workspace and Status parameters and the entire Message Buffer are internal, private registers used by the ECOM100 Configuration IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

In order for MOST ECOM100 IBoxes to function, you must turn ON dip switch 7 on the ECOM100 circuit board. You can keep dip switch 7 off if you are ONLY using ECOM100 Network Read and Write IBoxes (ECRX, ECWX).

ECOM100 Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Slot: specifies which PLC slot is occupied by the ECOM100 module
- Status: specifies a V-memory location that will be used by the instruction
- Workspace: specifies a V-memory location that will be used by the instruction
- Msg Buffer: specifies the starting address of a 65 word buffer that will be used by the module for configuration

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Slot . . . . K	K0-7
Status V	See DL405 V-memory map - Data Words
Workspace V	See DL405 V-memory map - Data Words
Msg Buffer (65 words used) V	See DL405 V-memory map - Data Words

ECOM100 Example

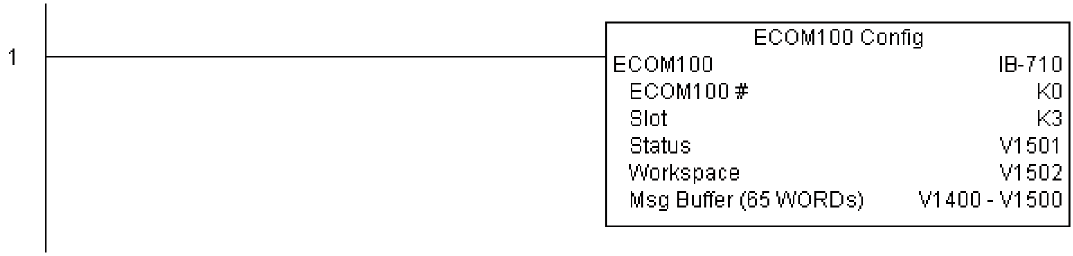
The ECOM100 Config IBox coordinates all of the interaction with other ECOM100 based IBoxes (ECxxxx). You must have an ECOM100 Config IBox for each ECOM100 module in your system. Configuration IBoxes must be at the top of your program and must execute every scan.

This IBox defines ECOM100# K0 to be in slot 3. Any ECOM100 IBoxes that need to reference this specific module (such as ECEMAIL, ECRX, ...) would enter K0 for their ECOM100# parameter.

The Status register is for reporting any completion or error information to other ECOM100 IBoxes. This V memory register must not be used anywhere else in the entire program.

The Workspace register is used to maintain state information about the ECOM100, along with proper sharing and interlocking with the other ECOM100 IBoxes in the program. This V memory register must not be used anywhere else in the entire program.

The Message Buffer of 65 words (130 bytes) is a common pool of memory that is used by other ECOM100 IBoxes (such as ECEMAIL). This way, you can have a bunch of ECEMAIL IBoxes, but only need 1 common buffer for generating and sending each EMail. These V memory registers must not be used anywhere else in your entire program.

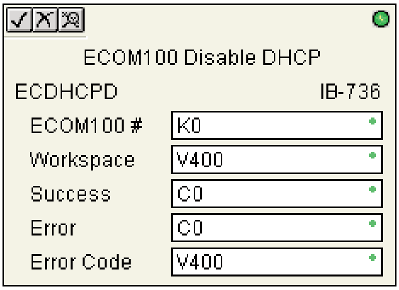


```
1                                   ECOM100 Config
                         ECOM100                          IB-710
                           ECOM100 #                          K0
                           Slot                               K3
                           Status                 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

ECOM100 Disable DHCP (ECDHCPD) (IB-736)

DS5/6	Used
HPP	N/A

ECOM100 Disable DHCP will setup the ECOM100 to use its internal TCP/IP settings on a leading edge transition to the IBox. To configure the ECOM100's TCP/IP settings manually, use the NetEdit3 utility, or you can do it programmatically from your PLC program using the ECOM100 IP Setup (ECIPSUP), or the individual ECOM100 IBoxes: ECOM Write IP Address (ECWRIP), ECOM Write Gateway Address (ECWRGWA), and ECOM100 Write Subnet Mask (ECWRSNM).


ECOM100 Disable DHCP
ECDHCPD IB-736
ECOM100 # K0
Workspace V400
Success C0
Error C0
Error Code V400

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete. If there is an error, the Error Code parameter will report an ECOM100 error code (less than 100), or a PLC logic error (greater than 1000).

The "Disable DHCP" setting is stored in Flash-ROM in the ECOM100 and the execution of this IBox will disable the ECOM100 module for at least a half second until it writes the Flash-ROM. Therefore, it is HIGHLY RECOMMENDED that you only execute this IBox ONCE, on the second scan. Since it requires a LEADING edge to execute, use a NORMALLY CLOSED SP0 (STR NOT First Scan) to drive the power flow to the IBox.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

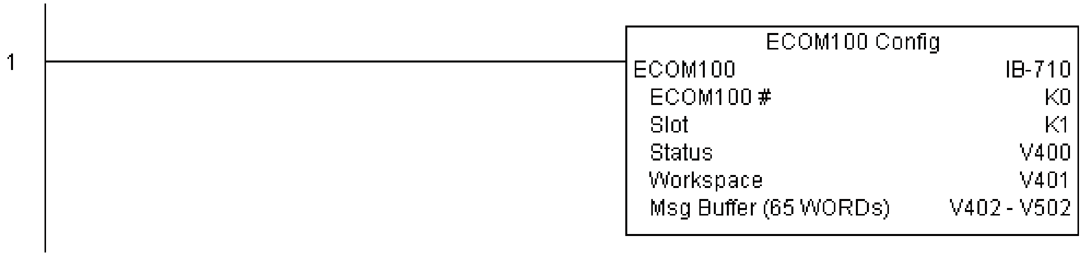
ECDHCPD Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Error Code: specifies the location where the Error Code will be written

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error Code V	See DL405 V-memory map - Data Words

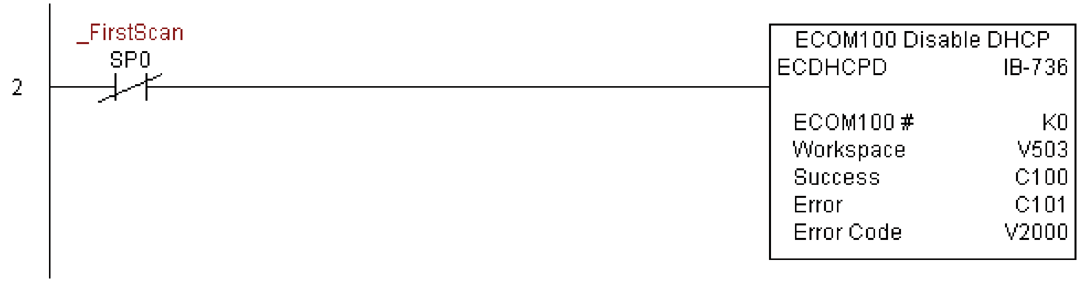
ECDHCPD Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module. V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



```
ECOM100 Config
ECOM100                        IB-710
  ECOM100 #                        K0
  Slot                             K1
  Status                         V400
  Workspace                      V401
  Msg Buffer (65 WORDs)   V402 - V502
```

Rung 2: On the 2nd scan, disable DHCP in the ECOM100. DHCP is the same protocol used by PCs for using a DHCP Server to automatically assign the ECOM100's IP Address, Gateway Address, and Subnet Mask. Typically disabling DHCP is done by assigning a hard-coded IP Address either in NetEdit or using one of the ECOM100 IP Setup IBoxes, but this IBox allows you to disable DHCP in the ECOM100 using your ladder program. The ECDHCPD is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to disable DHCP will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON. If successful, turn on C100. If there is a failure, turn on C101. If it fails, you can look at V2000 for the specific error code.

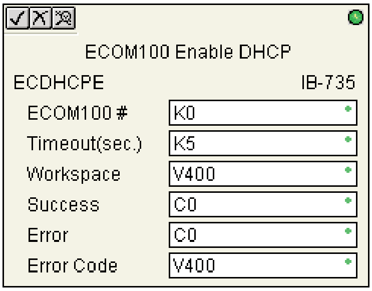


```
ECOM100 Disable DHCP
ECDHCPD             IB-736

  ECOM100 #             K0
  Workspace           V503
  Success             C100
  Error               C101
  Error Code         V2000
```

ECOM100 Enable DHCP (ECDHCPE) (IB-735)

DS5/6	Used
HPP	N/A

ECOM100 Enable DHCP will tell the ECOM100 to obtain its TCP/IP setup from a DHCP Server on a leading edge transition to the IBox.



ECOM100 Enable DHCP

ECDHCPE IB-735

ECOM100 #	K0
Timeout(sec.)	K5
Workspace	V400
Success	C0
Error	C0
Error Code	V400

The IBox will be successful once the ECOM100 has received its TCP/IP settings from the DHCP server. Since it is possible for the DHCP server to be unavailable, a Timeout parameter is provided so that the IBox can complete, but with an Error (Error Code = 1004 decimal).

See also the ECOM100 IP Setup (ECIPSUP) IBox 717 to directly setup ALL of the TCP/IP parameters in a single instruction - IP Address, Subnet Mask, and Gateway Address.

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete. If there is an error, the Error Code parameter will report an ECOM100 error code (less than 100), or a PLC logic error (greater than 1000).

The "Enable DHCP" setting is stored in Flash-ROM in the ECOM100 and the execution of this IBox will disable the ECOM100 module for at least a half second until it writes the Flash-ROM. Therefore, it is HIGHLY RECOMMENDED that you only execute this IBox ONCE, on the second scan. Since it requires a LEADING edge to execute, use a NORMALLY CLOSED SP0 (STR NOT First Scan) to drive the power flow to the IBox.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

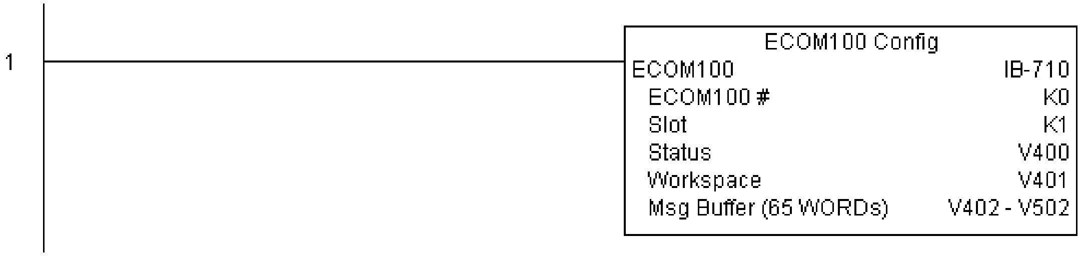
ECDHCPE Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Timeout(sec): specifies a timeout period so that the instruction may have time to complete
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Error Code: specifies the location where the Error Code will be written

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Timeout (sec) . . . . K	K5-127
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error Code V	See DL405 V-memory map - Data Words

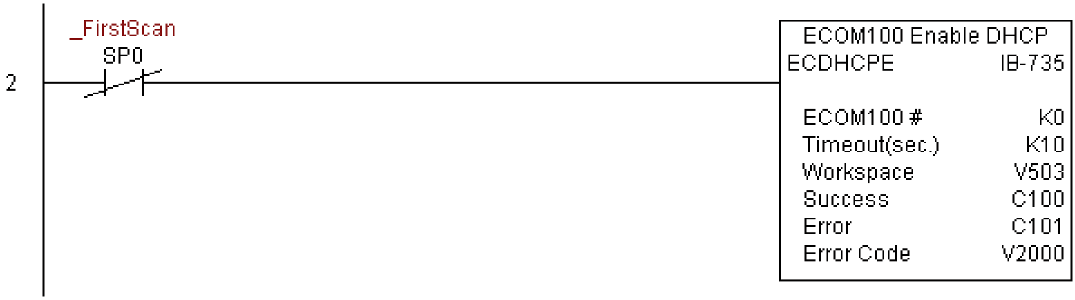
ECDHCPE Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



```
ECOM100 Config
ECOM100                          IB-710
  ECOM100 #                          K0
  Slot                               K1
  Status                           V400
  Workspace                        V401
  Msg Buffer (65 WORDs)     V402 - V502
```

Rung 2: On the 2nd scan, enable DHCP in the ECOM100. DHCP is the same protocol used by PCs for using a DHCP Server to automatically assign the ECOM100's IP Address, Gateway Address, and Subnet Mask. Typically this is done using NetEdit, but this IBox allows you to enable DHCP in the ECOM100 using your ladder program. The ECDHCPE is leading edge triggered, not power-flow driven (similar to a counter input leg). The commands to enable DHCP will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON. The ECDHCPE does more than just set the bit to enable DHCP in the ECOM100, but it then polls the ECOM100 once every second to see if the ECOM100 has found a DHCP server and has a valid IP Address. Therefore, a timeout parameter is needed in case the ECOM100 cannot find a DHCP server. If a timeout does occur, the Error bit will turn on and the error code will be 1004 decimal. The Success bit will turn on only if the ECOM100 finds a DHCP Server and is assigned a valid IP Address. If successful, turn on C100. If there is a failure, turn on C101. If it fails, you can look at V2000 for the specific error code.

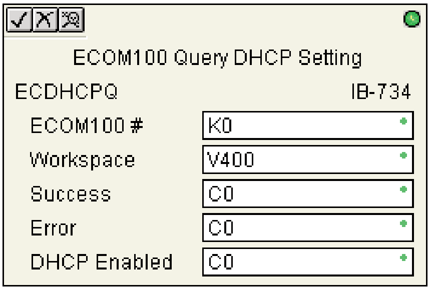


```
ECOM100 Enable DHCP
ECDHCPE              IB-735

  ECOM100 #              K0
  Timeout(sec.)         K10
  Workspace            V503
  Success              C100
  Error                C101
  Error Code          V2000
```

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

DS5/6	Used
HPP	N/A

ECOM100 Query DHCP Setting will determine if DHCP is enabled in the ECOM100 on a leading edge transition to the IBox. The DHCP Enabled bit parameter will be ON if DHCP is enabled, OFF if disabled.



ECOM100 Query DHCP Setting
ECDHCPQ IB-734
ECOM100 # K0
Workspace V400
Success C0
Error C0
DHCP Enabled C0

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

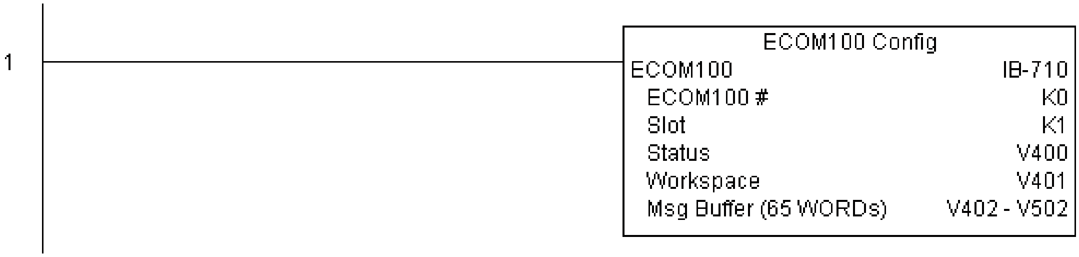
ECDHCPQ Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the instruction is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- DHCP Enabled: specifies a bit that will turn on if the ECOM100’s DHCP is enabled or remain off if disabled - after instruction query, be sure to check the state of the Success/Error bit state along with DHCP Enabled bit state to confirm a successful module query

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
DHCP Enabled . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

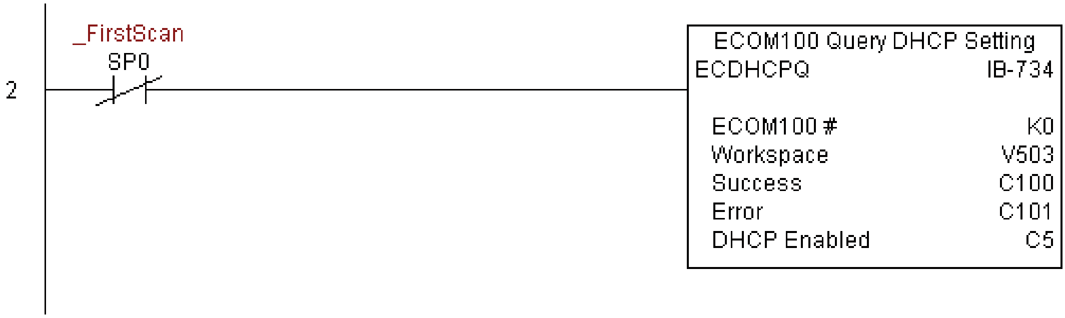
ECDHCPQ Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



```
ECOM100 Config
ECOM100                      IB-710
  ECOM100 #                      K0
  Slot                           K1
  Status                       V400
  Workspace                    V401
  Msg Buffer (65 WORDs)  V402 - V502
```

Rung 2: On the 2nd scan, read whether DHCP is enabled or disabled in the ECOM100 and store it in C5. DHCP is the same protocol used by PCs for using a DHCP Server to automatically assign the ECOM100's IP Address, Gateway Address, and Subnet Mask. The ECDHCPQ is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to read (Query) whether DHCP is enabled or not will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON. If successful, turn on C100. If there is a failure, turn on C101.

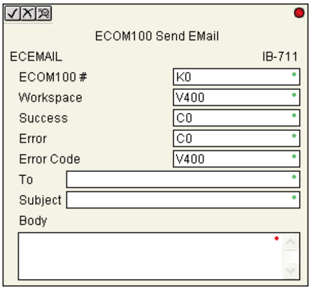


```
ECOM100 Query DHCP Setting
ECDHCPQ                  IB-734

  ECOM100 #                  K0
  Workspace                V503
  Success                  C100
  Error                    C101
  DHCP Enabled               C5
```

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

DS5/6	Used
HPP	N/A

ECOM100 Send EMail, on a leading edge transition, will behave as an EMail client and send an SMTP request to your SMTP Server to send the EMail message to the EMail addresses in the To: field and also to those listed in the Cc: list hard coded in the ECOM100. It will send the SMTP request based on the specified ECOM100#, which corresponds to a specific unique ECOM100 Configuration (ECOM100) at the top of your program.



ECOM100 Send EMail

ECEMAIL IB-711

ECOM100 #	K0
Workspace	V400
Success	C0
Error	C0
Error Code	V400
To	
Subject	
Body	

The Body: field supports what the PRINT and VPRINT instructions support for text and embedded variables, allowing you to embed real-time data in your EMail (e.g. "V2000 = " V2000:B).

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the request is complete. If there is an error, the Error Code parameter will report an ECOM100 error code (less than 100), an SMPT protocol error (between 100 and 999), or a PLC logic error (greater than 1000).

Since the ECOM100 is only an EMail Client and requires access to an SMTP Server, you MUST have the SMTP parameters configured properly in the ECOM100 via the ECOM100's Home Page and/or the EMail Setup instruction (ECEMSUP). To get to the ECOM100's Home Page, use your favorite Internet browser and browse to the ECOM100's IP Address, e.g. http://192.168.12.86

You are limited to approximately 100 characters of message data for the entire instruction, including the To: Subject: and Body: fields. To save space, the ECOM100 supports a hard coded list of EMail addresses for the Carbon Copy field (cc:) so that you can configure those IN the ECOM100, and keep the To: field small (or even empty), to leave more room for the Subject: and Body: fields.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

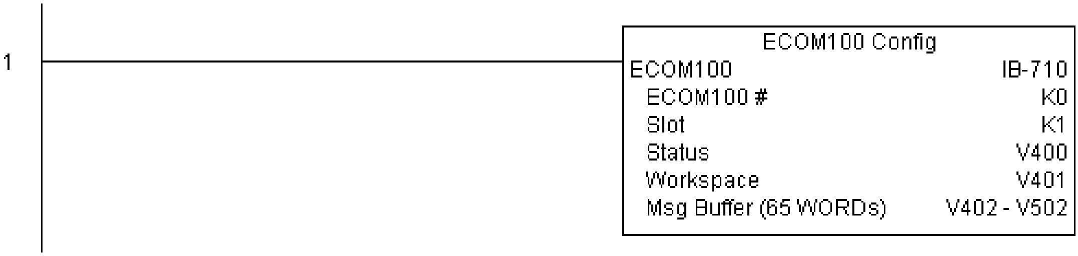
ECEMAIL Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Error Code: specifies the location where the Error Code will be written
- To: specifies an E-mail address that the message will be sent to
- Subject: subject of the e-mail message
- Body: supports what the PRINT and VPRINT instructions support for text and embedded variables, allowing you to embed real-time data in the EMail message

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error Code V	See DL405 V-memory map
To:	Text
Subject:	Text
Body:	See PRINT and VPRINT instructions

ECEMAIL Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



ECOM100 Config

ECOM100	IB-710
ECOM100 #	K0
Slot	K1
Status	V400
Workspace	V401
Msg Buffer (65 WORDs)	V402 - V502

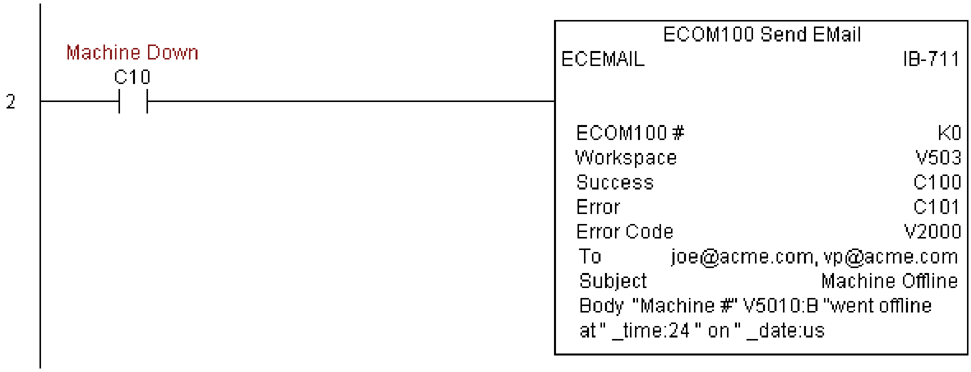
(example continued on next page)

ECEMAIL Example (con't)

Rung 2: When a machine goes down, send an email to Joe in maintenance and to the VP over production showing what machine is down along with the date/time stamp of when it went down.

The ECEMAIL is leading edge triggered, not power-flow driven (similar to a counter input leg). An email will be sent whenever the power flow into the IBox goes from OFF to ON. This helps prevent self inflicted spamming.

If the EMail is sent, turn on C100. If there is a failure, turn on C101. If it fails, you can look at V2000 for the SMTP error code or other possible error codes.



ECOM100 Send EMail
ECEMAIL IB-711

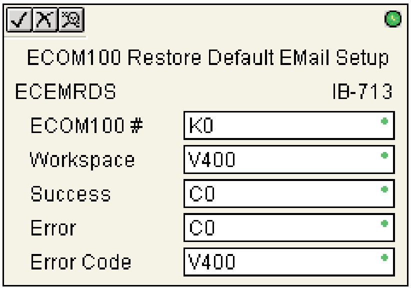
ECOM100 # K0
Workspace V503
Success C100
Error C101
Error Code V2000
To joe@acme.com, vp@acme.com
Subject Machine Offline
Body "Machine #" V5010:B "went offline at " _time:24 " on " _date:us

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

DS5/6	Used
HPP	N/A

ECOM100 Restore Default EMail Setup, on a leading edge transition, will restore the original EMail Setup data stored in the ECOM100 back to the working copy based on the specified ECOM100#, which corresponds to a specific unique ECOM100 Configuration (ECOM100) at the top of your program.

When the ECOM100 is first powered up, it copies the EMail setup data stored in ROM to the working copy in RAM. You can then modify this working copy from your program using the ECOM100 EMail Setup (ECEMSUP) IBox. After modifying the working copy, you can later restore the original setup data via your program by using this IBox.



ECOM100 Restore Default EMail Setup

ECEMRDS IB-713

ECOM100 #	K0
Workspace	V400
Success	C0
Error	C0
Error Code	V400

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete. If there is an error, the Error Code parameter will report an ECOM100 error code (less than 100), or a PLC logic error (greater than 1000).

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

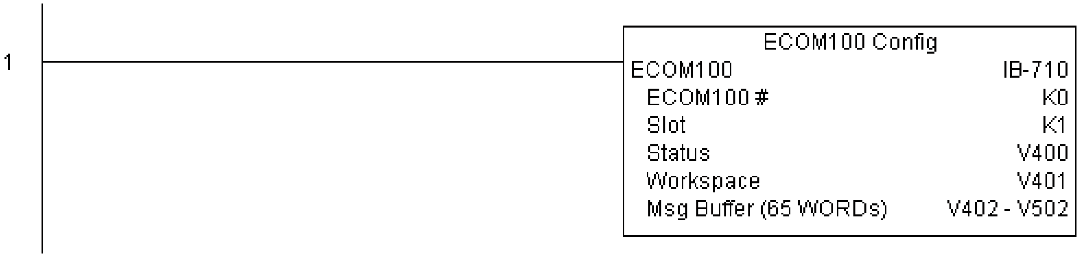
ECEMRDS Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Error Code: specifies the location where the Error Code will be written

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error Code V	See DL405 V-memory map - Data Words

ECEMRDS Example

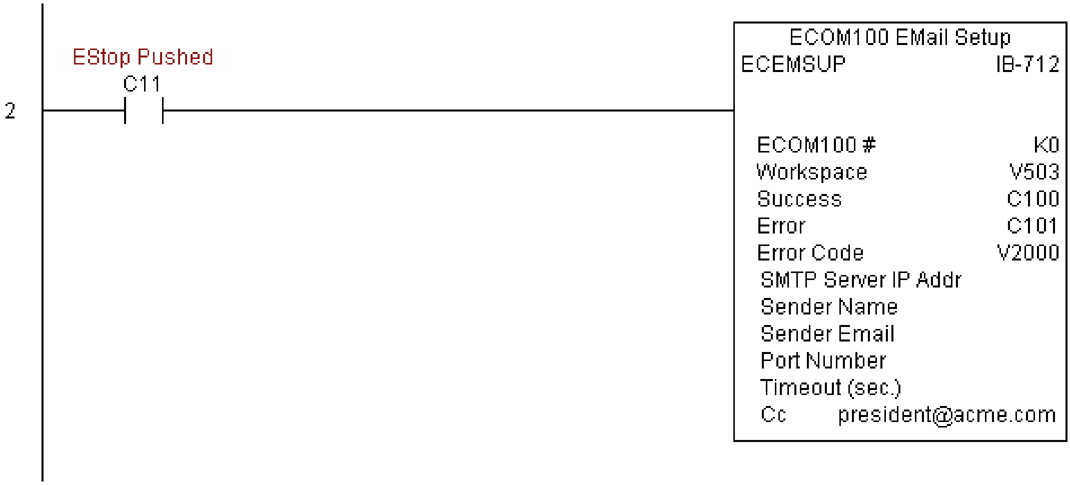
Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



```
                ECOM100 Config
ECOM100                           IB-710
  ECOM100 #                           K0
  Slot                                K1
  Status                            V400
  Workspace                         V401
  Msg Buffer (65 WORDs)      V402 - V502
```

Rung 2: Whenever an EStop is pushed, ensure that president of the company gets copies of all EMails being sent.

The ECOM100 EMail Setup IBox allows you to set/change the SMTP EMail settings stored in the ECOM100.



```
   ECOM100 EMail Setup
ECEMSUP                    IB-712


  ECOM100 #                    K0
  Workspace                  V503
  Success                    C100
  Error                      C101
  Error Code                V2000
  SMTP Server IP Addr
  Sender Name
  Sender Email
  Port Number
  Timeout (sec.)
  Cc       president@acme.com
```

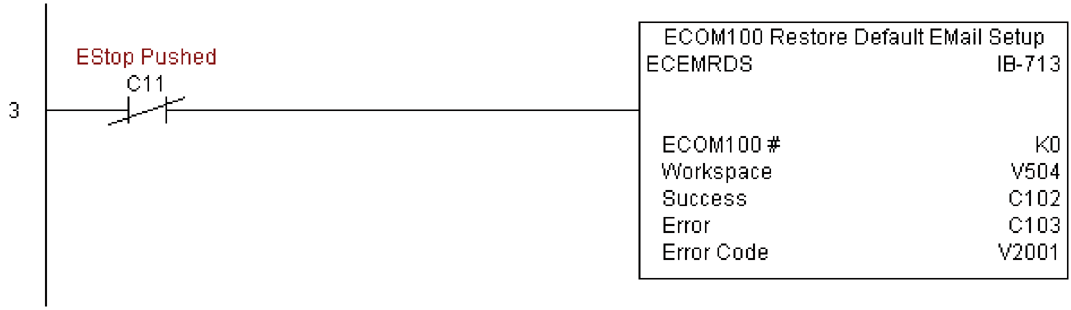
(example continued on next page)

ECEMRDS Example (con't)

Rung 3: Once the EStop is pulled out, take the president off the cc: list by restoring the default EMail setup in the ECOM100.

The ECEMRDS is leading edge triggered, not power-flow driven (similar to a counter input leg). The ROM based EMail configuration stored in the ECOM100 will be copied over the "working copy" whenever the power flow into the IBox goes from OFF to ON (the working copy can be changed by using the ECEMSUP IBox).

If successful, turn on C102. If there is a failure, turn on C103. If it fails, you can look at V2001 for the specific error code.



ECOM100 Restore Default EMail Setup
ECEMRDS IB-713

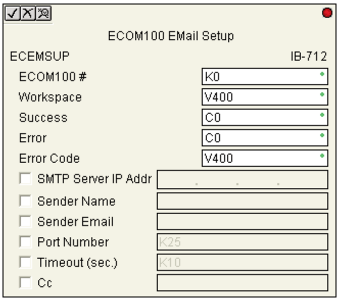
ECOM100 #	K0
Workspace	V504
Success	C102
Error	C103
Error Code	V2001

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

DS5/6	Used
HPP	N/A

ECOM100 EMail Setup, on a leading edge transition, will modify the working copy of the EMail setup currently in the ECOM100 based on the specified ECOM100#, which corresponds to a specific unique ECOM100 Configuration (ECOM100) at the top of your program.

You may pick and choose any or all fields to be modified using this instruction. Note that these changes are cumulative: if you execute multiple ECOM100 EMail Setup IBoxes, then all of the changes are made in the order they are executed. Also note that you can restore the original ECOM100 EMail Setup that is stored in the ECOM100 to the working copy by using the ECOM100 Restore Default EMail Setup (ECEMRDS) IBox.



ECOM100 EMail Setup

ECEMSUP IB-712

ECOM100 # K0

Workspace V400

Success C0

Error C0

Error Code V400

□ SMTP Server IP Addr . . .

□ Sender Name

□ Sender Email

□ Port Number K25

□ Timeout (sec.) K10

□ Cc

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete. If there is an error, the Error Code parameter will report an ECOM100 error code (less than 100), or a PLC logic error (greater than 1000).

You are limited to approximately 100 characters/bytes of setup data for the entire instruction. So if needed, you could divide the entire setup across multiple ECEMSUP IBoxes on a field-by-field basis, for example do the Carbon Copy (cc:) field in one ECEMSUP IBox and the remaining setup parameters in another.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

ECEMSUP Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Error Code: specifies the location where the Error Code will be written
- SMTP Server IP Addr: optional parameter that specifies the IP Address of the SMTP Server on the ECOM100's network
- Sender Name: optional parameter that specifies the sender name that will appear in the "From:" field to those who receive the e-mail
- Sender EMail: optional parameter that specifies the sender EMail address that will appear in the "From:" field to those who receive the e-mail

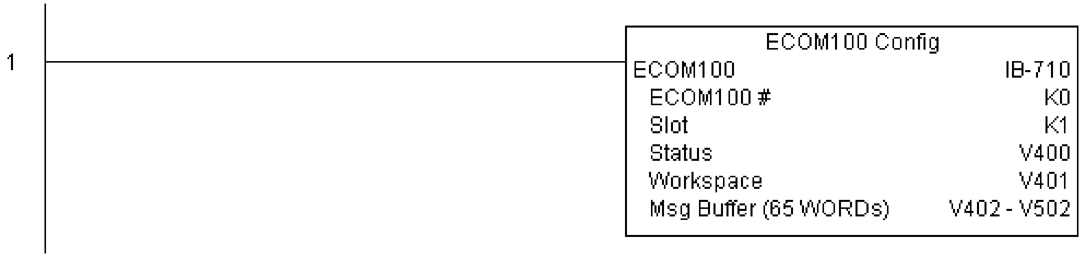
ECEMSUP Parameters (con't)

- Port Number: optional parameter that specifies the TCP/IP Port Number to send SMTP requests; usually this does not to be configured (see your network administrator for information on this setting)
- Timeout (sec): optional parameter that specifies the number of seconds to wait for the SMTP Server to send the EMail to all the recipients
- Cc: optional parameter that specifies a list of "carbon copy" Email addresses to send all EMails to

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error Code V	See DL405 V-memory map - Data Words

ECEMSUP Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



ECOM100 Config

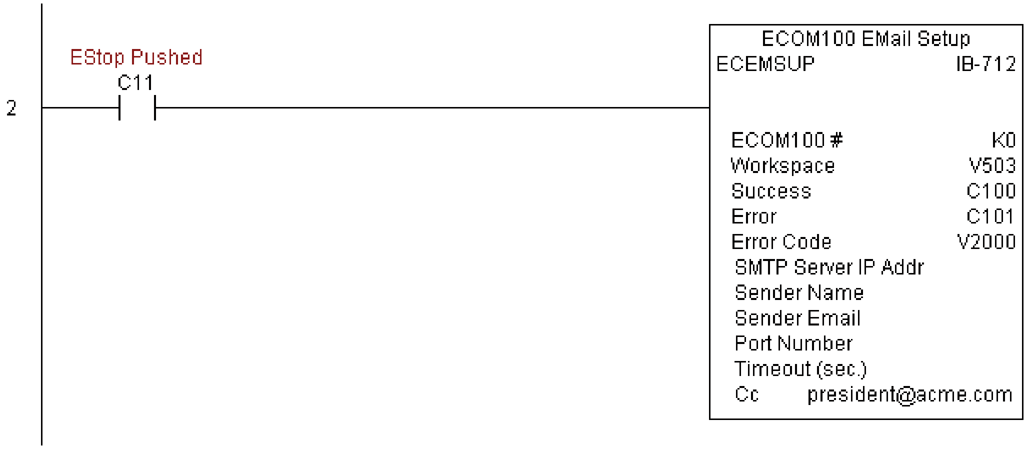
ECOM100	IB-710
ECOM100 #	K0
Slot	K1
Status	V400
Workspace	V401
Msg Buffer (65 WORDs)	V402 - V502

(example continued on next page)

ECEMSUP Example (con't)

Rung 2: Whenever an EStop is pushed, ensure that president of the company gets copies of all EMails being sent.The ECOM100 EMail Setup IBox allows you to set/change the SMTP EMail settings stored in the ECOM100. The ECEMSUP is leading edge triggered, not power-flow driven (similar to a counter input leg). At power-up, the ROM based EMail configuration stored in the ECOM100 is copied to a RAM based "working copy". You can change this working copy by using the ECEMSUP IBox. To restore the original ROM based configuration, use the Restore Default EMail Setup ECEMRDS IBox.

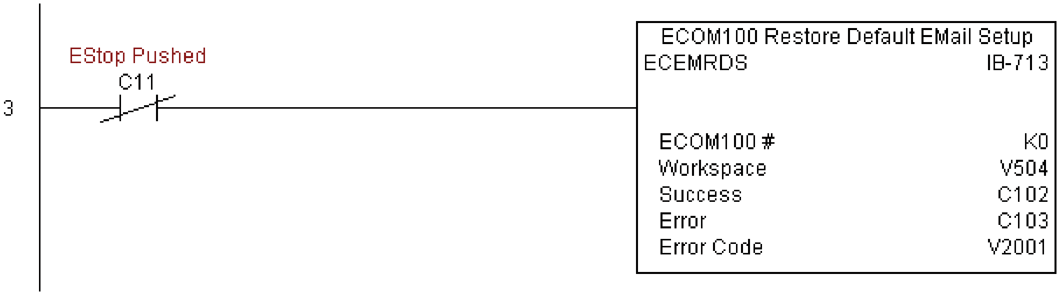
If successful, turn on C100. If there is a failure, turn on C101. If it fails, you can look at V2000 for the specific error code.



ECOM100 EMail Setup
ECEMSUP IB-712

ECOM100 # K0
Workspace V503
Success C100
Error C101
Error Code V2000
SMTP Server IP Addr
Sender Name
Sender Email
Port Number
Timeout (sec.)
Cc president@acme.com

Rung 3: Once the EStop is pulled out, take the president off the cc: list by restoring the default EMail setup in the ECOM100.



ECOM100 Restore Default EMail Setup
ECEMRDS IB-713

ECOM100 # K0
Workspace V504
Success C102
Error C103
Error Code V2001

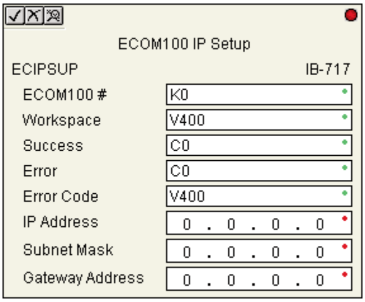
ECOM100 IP Setup (ECIPSUP) (IB-717)

DS5/6	Used
HPP	N/A

ECOM100 IP Setup will configure the three TCP/IP parameters in the ECOM100: IP Address, Subnet Mask, and Gateway Address, on a leading edge transition to the IBox. The ECOM100 is specified by the ECOM100#, which corresponds to a specific unique ECOM100 Configuration (ECOM100) IBox at the top of your program.

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete. If there is an error, the Error Code parameter will report an ECOM100 error code (less than 100), or a PLC logic error (greater than 1000).



ECOM100 IP Setup

ECIPSUP IB-717

ECOM100 #	K0
Workspace	V400
Success	C0
Error	C0
Error Code	V400
IP Address	0 . 0 . 0 . 0
Subnet Mask	0 . 0 . 0 . 0
Gateway Address	0 . 0 . 0 . 0

This setup data is stored in Flash-ROM in the ECOM100 and will disable the ECOM100 module for at least a half second until it writes the Flash-ROM. Therefore, it is HIGHLY RECOMMENDED that you only execute this IBox ONCE on the second scan. Since it requires a LEADING edge to execute, use a NORMALLY CLOSED SP0 (NOT First Scan) to drive the power flow to the IBox.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

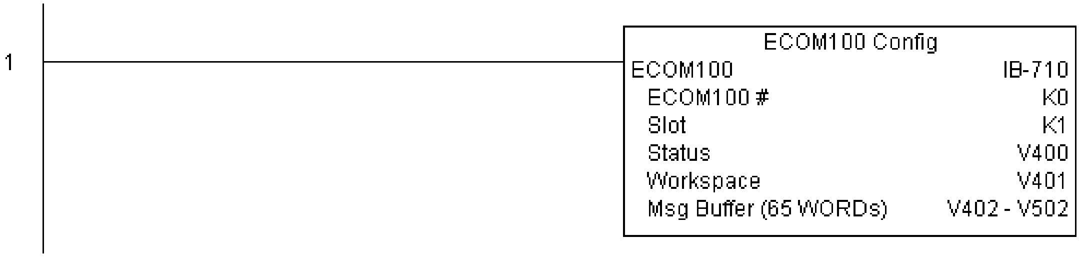
ECIPSUP Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Error Code: specifies the location where the Error Code will be written
- IP Address: specifies the module's IP Address
- Subnet Mask: specifies the Subnet Mask for the module to use
- Gateway Address: specifies the Gateway Address for the module to use

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error Code V	See DL405 V-memory map - Data Words
IP Address . . . . IP Address	0.0.0.1. to 255.255.255.254
Subnet Mask Address . . . . IP Address Mask	0.0.0.1. to 255.255.255.254
Gateway Address . . . . IP Address	0.0.0.1. to 255.255.255.254

ECIPSUP Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



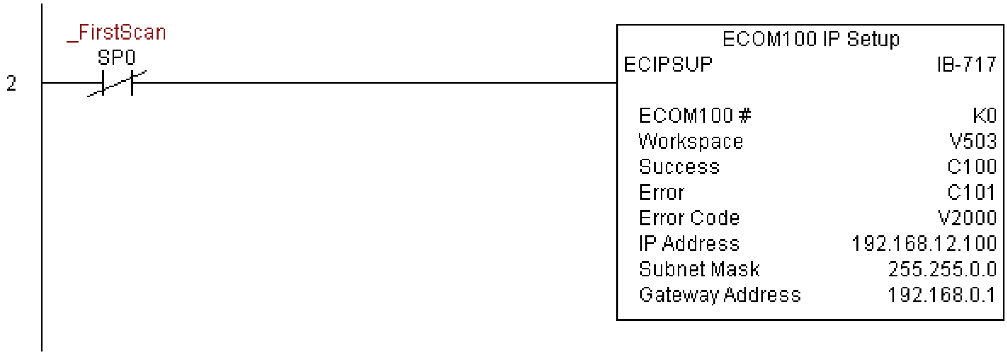
```
                ECOM100 Config
ECOM100                                  IB-710
  ECOM100 #                                  K0
  Slot                                       K1
  Status                                   V400
  Workspace                                V401
  Msg Buffer (65 WORDs)             V402 - V502
```

Rung 2: On the 2nd scan, configure all of the TCP/IP parameters in the ECOM100:

IP Address:	192.168. 12.100
Subnet Mask:	255.255. 0. 0
Gateway Address:	192.168. 0. 1

The ECIPSUP is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to write the TCP/IP configuration parameters will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON.

If successful, turn on C100. If there is a failure, turn on C101. If it fails, you can look at V2000 for the specific error code.



```
             ECOM100 IP Setup
ECIPSUP                               IB-717

  ECOM100 #                               K0
  Workspace                             V503
  Success                               C100
  Error                                 C101
  Error Code                           V2000
  IP Address                  192.168.12.100
  Subnet Mask                    255.255.0.0
  Gateway Address                192.168.0.1
```

ECOM100 Read Description (ECRDDES) (IB-726)

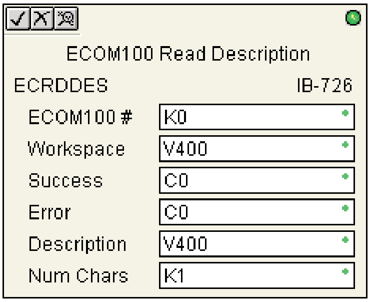
DS5/6	Used
HPP	N/A

ECOM100 Read Description will read the ECOM100's Description field up to the number of specified characters on a leading edge transition to the IBox.

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.


ECOM100 Read Description
ECRDDES IB-726
ECOM100 # K0
Workspace V400
Success C0
Error C0
Description V400
Num Chars K1

S

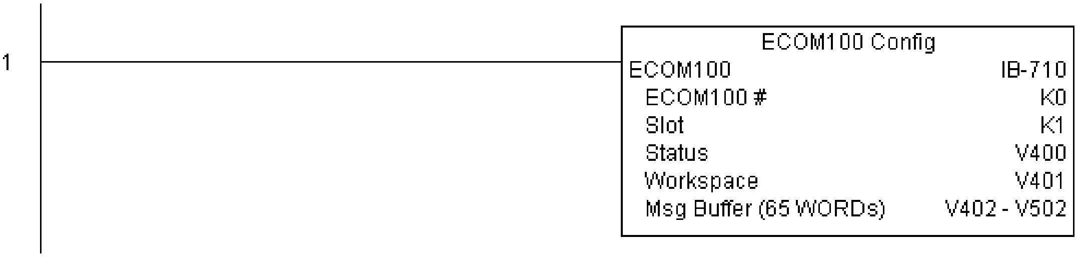
ECRDDES Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Description: specifies the starting buffer location where the ECOM100's Module Name will be placed
- Num Char: specifies the number of characters (bytes) to read from the ECOM100's Description field

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Description V	See DL405 V-memory map - Data Words
Num Chars . . . . K	K1-128

ECRDDES Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.

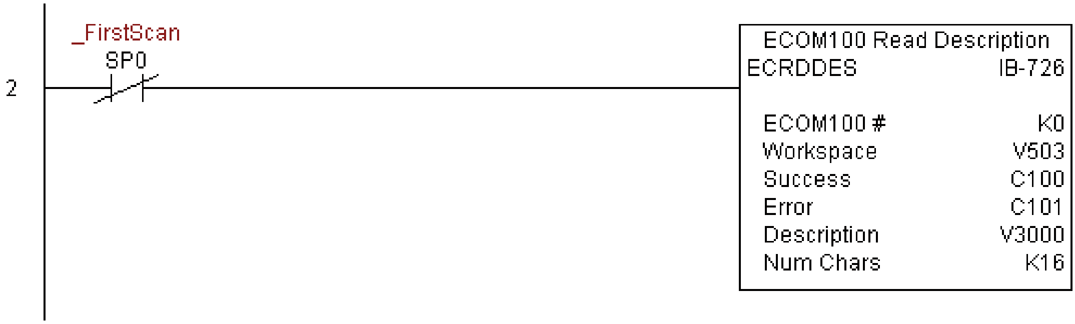


```
ECOM100 Config
ECOM100                      IB-710
  ECOM100 #                      K0
  Slot                           K1
  Status                       V400
  Workspace                    V401
  Msg Buffer (65 WORDs)  V402 - V502
```

Rung 2: On the 2nd scan, read the Module Description of the ECOM100 and store it in V3000 thru V3007 (16 characters). This text can be displayed by an HMI.

The ECRDDES is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to read the module description will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON.

If successful, turn on C100. If there is a failure, turn on C101.

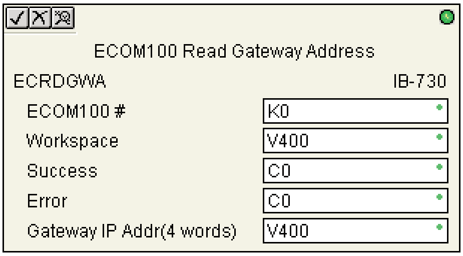


```
ECOM100 Read Description
ECRDDES                IB-726

  ECOM100 #                K0
  Workspace              V503
  Success                C100
  Error                  C101
  Description           V3000
  Num Chars               K16
```

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

DS5/6	Used
HPP	N/A

ECOM100 Read Gateway Address will read the 4 parts of the Gateway IP address and store them in 4 consecutive V Memory locations in decimal format, on a leading edge transition to the IBox.



The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

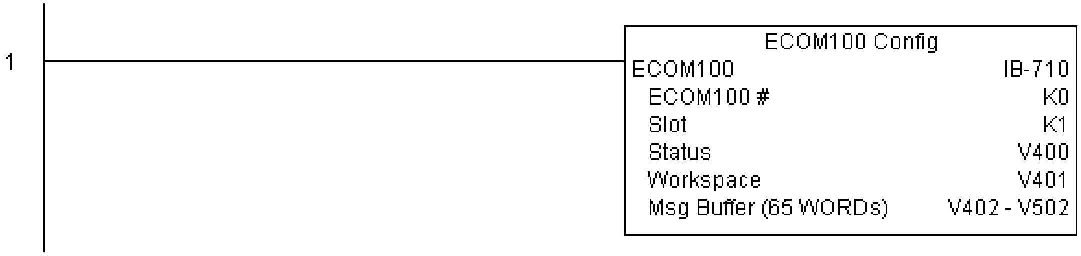
ECRDGWA Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Gateway IP Addr: specifies the starting address where the ECOM100's Gateway Address will be placed in 4 consecutive V-memory locations

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Gateway IP Address (4 Words) V	See DL405 V-memory map - Data Words

ECRDGWA Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.

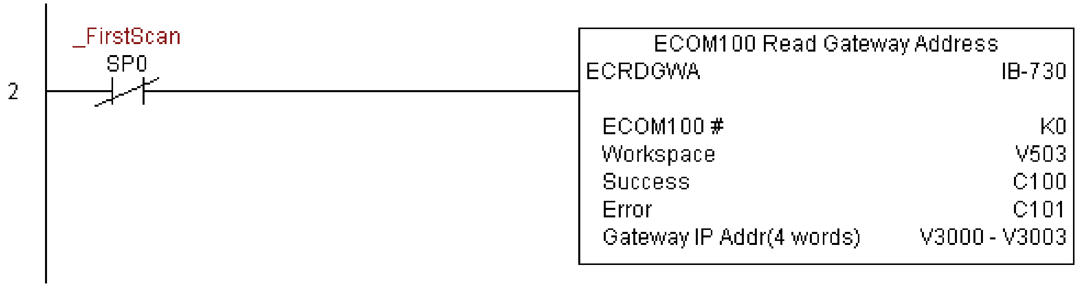


ECOM100 Config
ECOM100 IB-710
ECOM100 # K0
Slot K1
Status V400
Workspace V401
Msg Buffer (65 WORDs) V402 - V502

Rung 2: On the 2nd scan, read the Gateway Address of the ECOM100 and store it in V3000 thru V3003 (4 decimal numbers). The ECOM100's Gateway Address could be displayed by an HMI.

The ECRDGWA is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to read the Gateway Address will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON.

If successful, turn on C100. If there is a failure, turn on C101.



ECOM100 Read Gateway Address
ECRDGWA IB-730
ECOM100 # K0
Workspace V503
Success C100
Error C101
Gateway IP Addr(4 words) V3000 - V3003

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

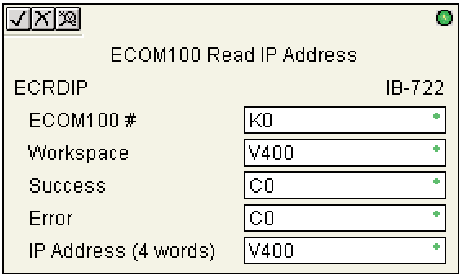
DS5/6	Used
HPP	N/A

ECOM100 Read IP Address will read the 4 parts of the IP address and store them in 4 consecutive V Memory locations in decimal format, on a leading edge transition to the IBox.

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.


ECOM100 Read IP Address

ECRDIP IB-722

ECOM100 # K0

Workspace V400

Success C0

Error C0

IP Address (4 words) V400

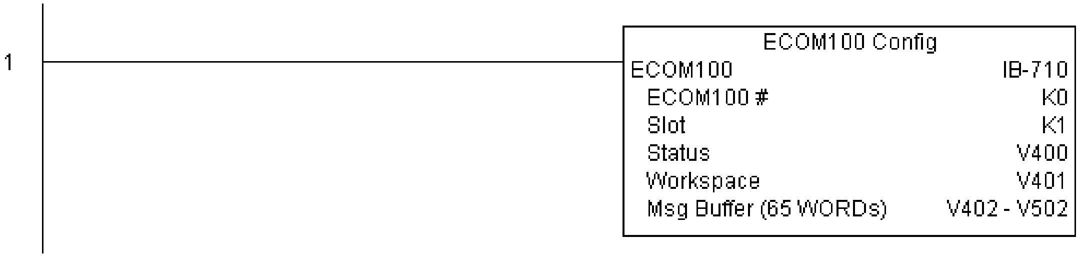
ECRDIP Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- IP Address: specifies the starting address where the ECOM100’s IP Address will be placed in 4 consecutive V-memory locations

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
IP Address (4 Words) V	See DL405 V-memory map - Data Words

ECRDIP Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.

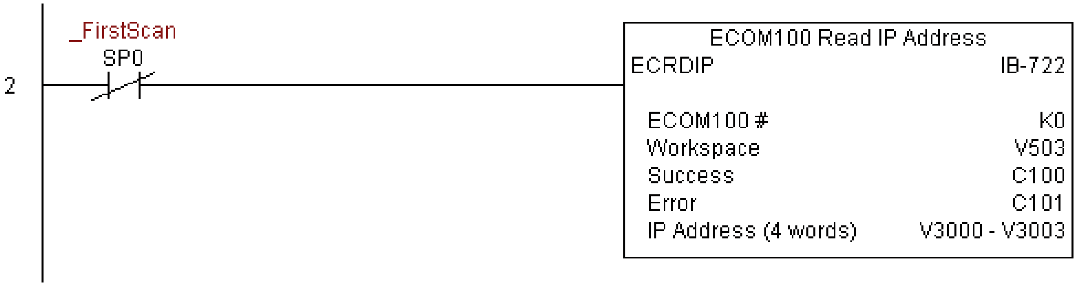


ECOM100 Config
ECOM100 IB-710
ECOM100 # K0
Slot K1
Status V400
Workspace V401
Msg Buffer (65 WORDs) V402 - V502

Rung 2: On the 2nd scan, read the IP Address of the ECOM100 and store it in V3000 thru V3003 (4 decimal numbers). The ECOM100's IP Address could be displayed by an HMI.

The ECRDIP is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to read the IP Address will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON.

If successful, turn on C100. If there is a failure, turn on C101.



ECOM100 Read IP Address
ECRDIP IB-722
ECOM100 # K0
Workspace V503
Success C100
Error C101
IP Address (4 words) V3000 - V3003

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

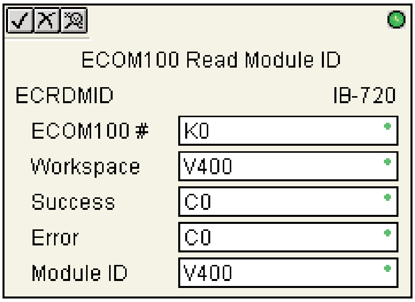
DS5/6	Used
HPP	N/A

ECOM100 Read Module ID will read the binary (decimal) WORD sized Module ID on a leading edge transition to the IBox.

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.



ECOM100 Read Module ID

ECRDMID IB-720

ECOM100 # K0

Workspace V400

Success C0

Error C0

Module ID V400

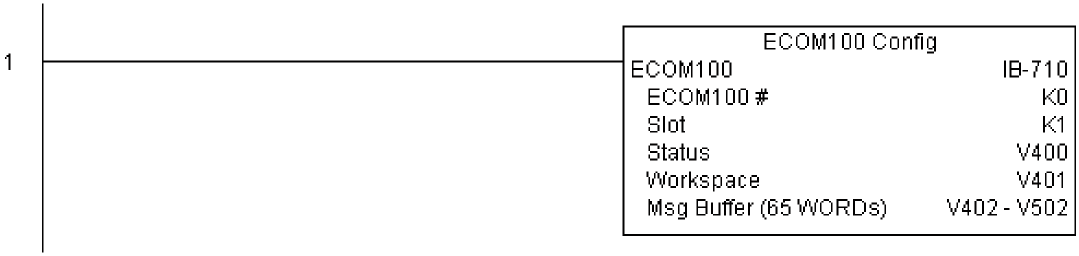
ECRDMID Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Module ID: specifies the location where the ECOM100’s Module ID (decimal) will be placed

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Module ID. V	See DL405 V-memory map - Data Words

ECRDMID Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



1

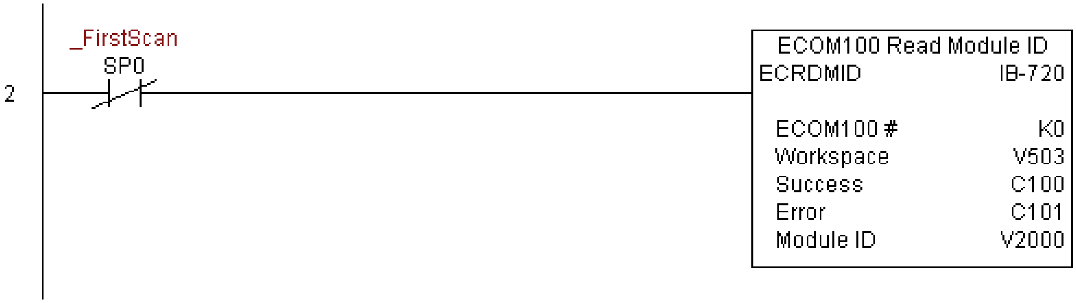
ECOM100 Config

ECOM100	IB-710
ECOM100 #	K0
Slot	K1
Status	V400
Workspace	V401
Msg Buffer (65 WORDs)	V402 - V502

Rung 2: On the 2nd scan, read the Module ID of the ECOM100 and store it in V2000.

The ECRDMID is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to read the module ID will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON.

If successful, turn on C100. If there is a failure, turn on C101.



2

_FirstScan
SP0

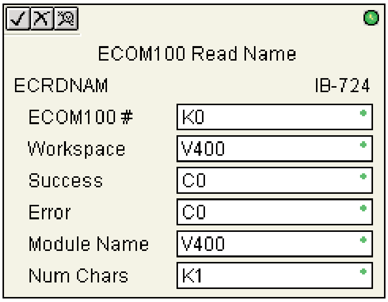
ECOM100 Read Module ID

ECRDMID	IB-720
ECOM100 #	K0
Workspace	V503
Success	C100
Error	C101
Module ID	V2000

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

DS5/6	Used
HPP	N/A

ECOM100 Read Name will read the Module Name up to the number of specified characters on a leading edge transition to the IBox.


ECOM100 Read Name
ECRDNAM IB-724
ECOM100 # K0
Workspace V400
Success C0
Error C0
Module Name V400
Num Chars K1

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

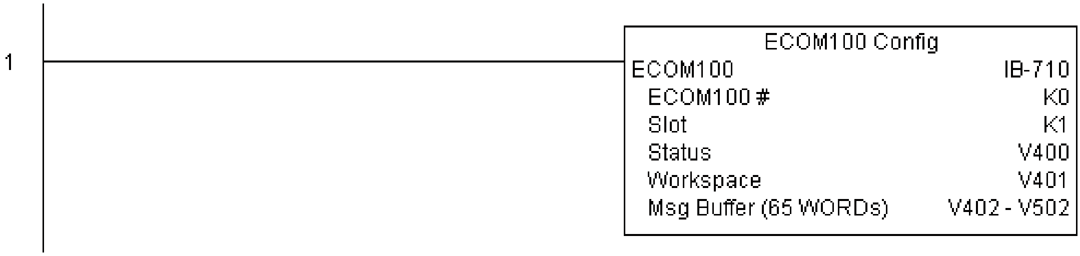
ECRDNAM Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Module Name: specifies the starting buffer location where the ECOM100's Module Name will be placed
- Num Chars: specifies the number of characters (bytes) to read from the ECOM100's Name field

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Module Name V	See DL405 V-memory map - Data Words
Num Chars . . . . K	K1-128

ECRDNAM Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.

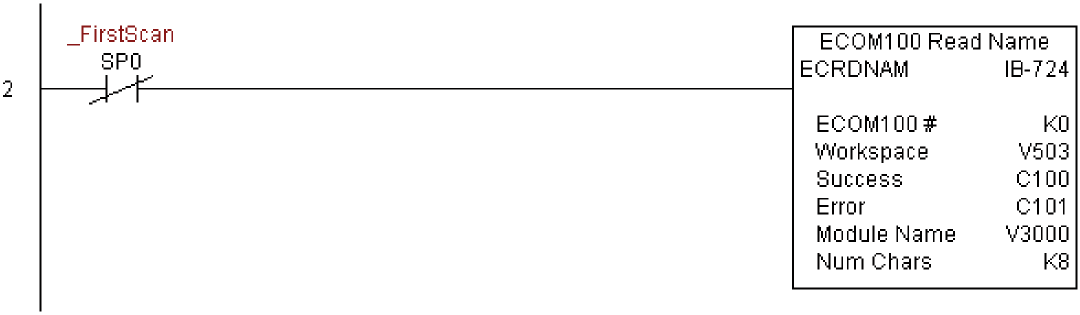


```
1
                         ECOM100 Config
ECOM100                                  IB-710
  ECOM100 #                                  K0
  Slot                                       K1
  Status                                   V400
  Workspace                                V401
  Msg Buffer (65 WORDs)             V402 - V502
```

Rung 2: On the 2nd scan, read the Module Name of the ECOM100 and store it in V3000 thru V3003 (8 characters). This text can be displayed by an HMI.

The ECRDNAM is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to read the module name will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON.

If successful, turn on C100. If there is a failure, turn on C101.

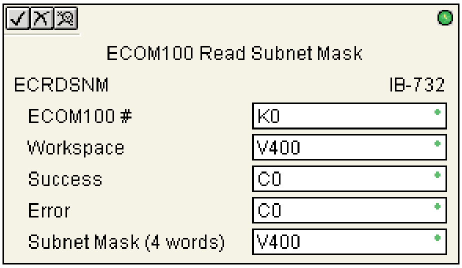


```
2   _FirstScan
    SP0
                 ECOM100 Read Name
               ECRDNAM          IB-724

                 ECOM100 #          K0
                 Workspace        V503
                 Success          C100
                 Error            C101
                 Module Name     V3000
                 Num Chars          K8
```

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

DS5/6	Used
HPP	N/A

ECOM100 Read Subnet Mask will read the 4 parts of the Subnet Mask and store them in 4 consecutive V Memory locations in decimal format, on a leading edge transition to the IBox.



ECOM100 Read Subnet Mask
ECRDSNM IB-732
ECOM100 #: K0
Workspace: V400
Success: C0
Error: C0
Subnet Mask (4 words): V400

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

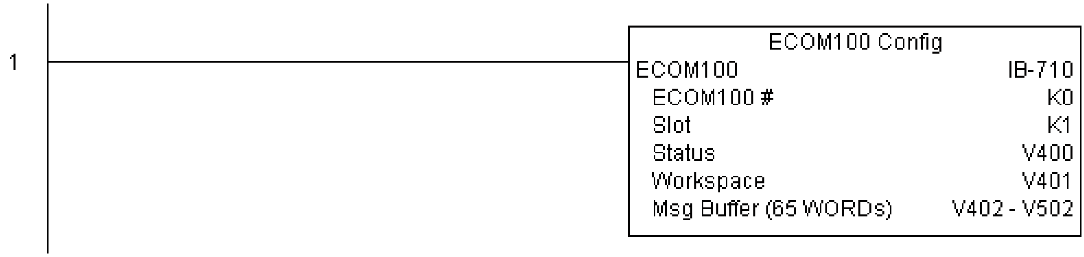
ECRDSNM Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Subnet Mask: specifies the starting address where the ECOM100's Subnet Mask will be placed in 4 consecutive V-memory locations

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Subnet Mask (4 Words). V	See DL405 V-memory map - Data Words

ECRDSNM Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.

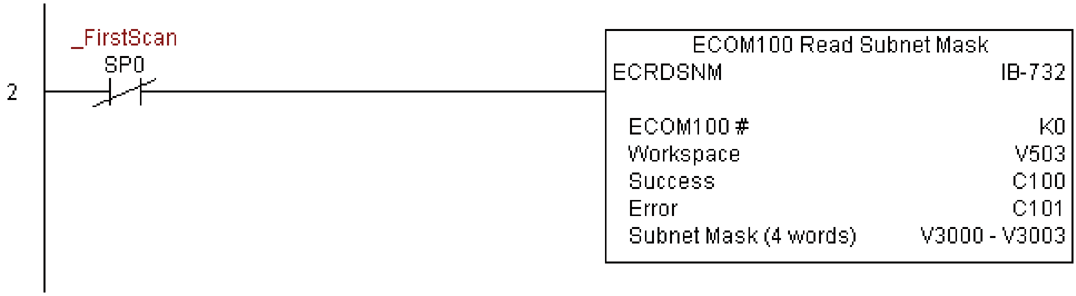


ECOM100 Config	
ECOM100	IB-710
ECOM100 #	K0
Slot	K1
Status	V400
Workspace	V401
Msg Buffer (65 WORDs)	V402 - V502

Rung 2: On the 2nd scan, read the Subnet Mask of the ECOM100 and store it in V3000 thru V3003 (4 decimal numbers). The ECOM100's Subnet Mask could be displayed by an HMI.

The ECRDSNM is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to read the Subnet Mask will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON.

If successful, turn on C100. If there is a failure, turn on C101.

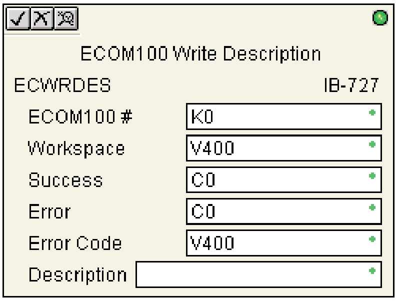


ECOM100 Read Subnet Mask	
ECRDSNM	IB-732
ECOM100 #	K0
Workspace	V503
Success	C100
Error	C101
Subnet Mask (4 words)	V3000 - V3003

ECOM100 Write Description (ECWRDES) (IB-727)

DS5/6	Used
HPP	N/A

ECOM100 Write Description will write the given Description to the ECOM100 module on a leading edge transition to the IBox. If you use a dollar sign ($) or double quote ("), use the PRINT/VPRINT escape sequence of TWO dollar signs ($$) for a single dollar sign or dollar sign-double quote ($") for a double quote character.



ECOM100 Write Description

ECWRDES IB-727

ECOM100 # K0

Workspace V400

Success C0

Error C0

Error Code V400

Description

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete. If there is an error, the Error Code parameter will report an ECOM100 error code (less than 100), or a PLC logic error (greater than 1000).

The Description is stored in Flash-ROM in the ECOM100 and the execution of this IBox will disable the ECOM100 module for at least a half second until it writes the Flash-ROM. Therefore, it is HIGHLY RECOMMENDED that you only execute this IBox ONCE on the second scan. Since it requires a LEADING edge to execute, use a NORMALLY CLOSED SP0 (STR NOT First Scan) to drive the power flow to the IBox.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

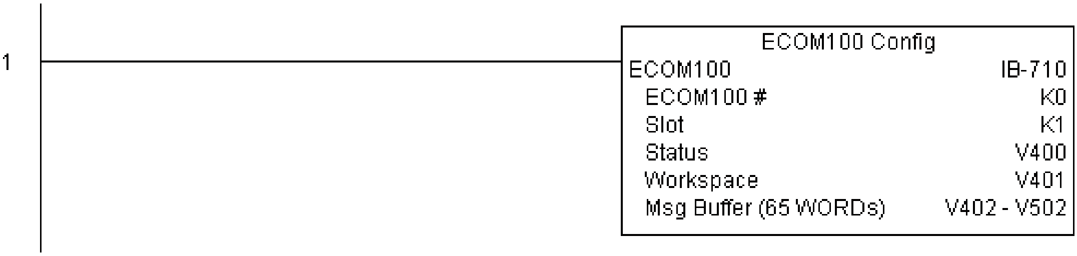
ECWRDES Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Error Code: specifies the location where the Error Code will be written
- Description: specifies the Description that will be written to the module

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error Code V	See DL405 V-memory map - Data Words
Description	Text

ECWRDES Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.

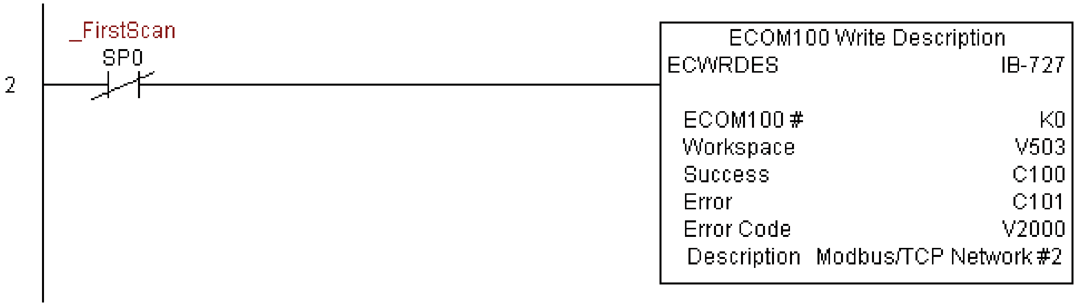


ECOM100 Config
ECOM100 IB-710
ECOM100 # K0
Slot K1
Status V400
Workspace V401
Msg Buffer (65 WORDs) V402 - V502

Rung 2: On the 2nd scan, set the Module Description of the ECOM100. Typically this is done using NetEdit, but this IBox allows you to configure the module description in the ECOM100 using your ladder program.

The EWRDES is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to write the module description will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON.

If successful, turn on C100. If there is a failure, turn on C101. If it fails, you can look at V2000 for the specific error code.

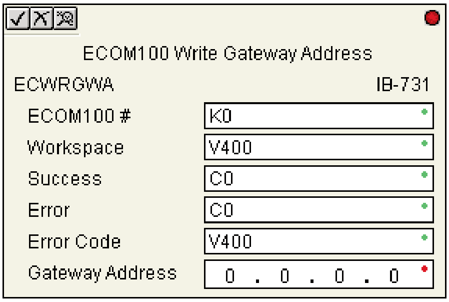


ECOM100 Write Description
ECWRDES IB-727
ECOM100 # K0
Workspace V503
Success C100
Error C101
Error Code V2000
Description Modbus/TCP Network #2

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

DS5/6	Used
HPP	N/A

ECOM100 Write Gateway Address will write the given Gateway IP Address to the ECOM100 module on a leading edge transition to the IBox. See also ECOM100 IP Setup (ECIPSUP) IBox 717 to setup ALL of the TCP/IP parameters in a single instruction - IP Address, Subnet Mask, and Gateway Address.



ECOM100 Write Gateway Address

ECWRGWA IB-731

ECOM100 # K0

Workspace V400

Success C0

Error C0

Error Code V400

Gateway Address 0 . 0 . 0 . 0

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete. If there is an error, the Error Code parameter will report an ECOM100 error code (less than 100), or a PLC logic error (greater than 1000).

The Gateway Address is stored in Flash-ROM in the ECOM100 and the execution of this IBox will disable the ECOM100 module for at least a half second until it writes the Flash-ROM. Therefore, it is HIGHLY RECOMMENDED that you only execute this IBox ONCE, on the second scan. Since it requires a LEADING edge to execute, use a NORMALLY CLOSED SP0 (STR NOT First Scan) to drive the power flow to the IBox.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

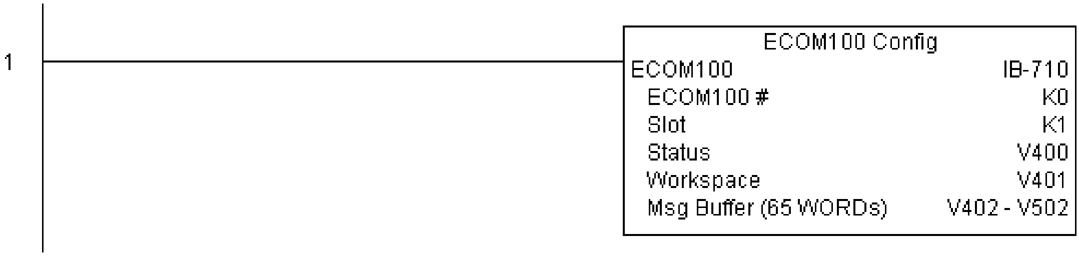
ECWRGWA Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Error Code: specifies the location where the Error Code will be written
- Gateway Address: specifies the Gateway IP Address that will be written to the module

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error Code V	See DL405 V-memory map - Data Words
Gateway Address	0.0.0.1. to 255.255.255.254

ECWRGWA Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



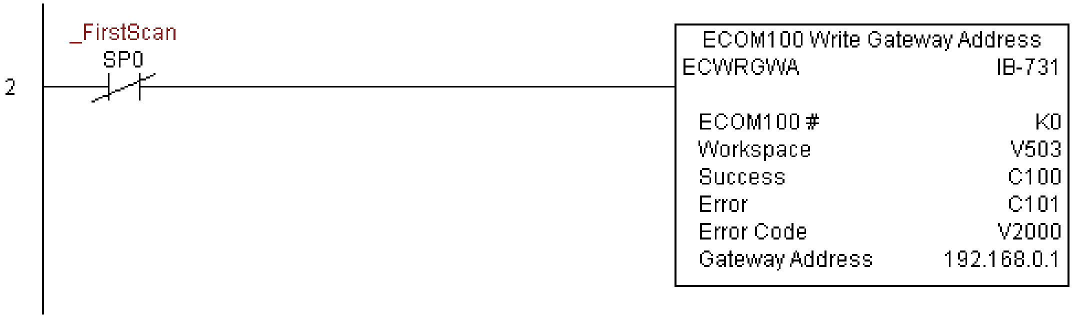
```
ECOM100 Config
ECOM100                      IB-710
  ECOM100 #                      K0
  Slot                           K1
  Status                       V400
  Workspace                    V401
  Msg Buffer (65 WORDs)  V402 - V502
```

Rung 2: On the 2nd scan, assign the Gateway Address of the ECOM100 to 192.168.0.1

The ECWRGWA is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to write the Gateway Address will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON.

If successful, turn on C100. If there is a failure, turn on C101. If it fails, you can look at V2000 for the specific error code.

To configure all of the ECOM100 TCP/IP parameters in one IBox, see the ECOM100 IP Setup (ECIPSUP) IBox.

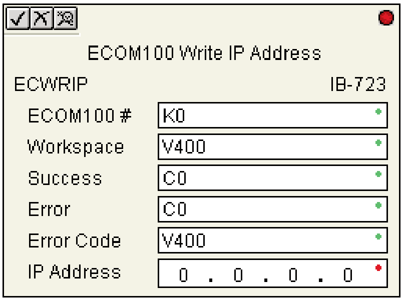


```
ECOM100 Write Gateway Address
ECWRGWA                      IB-731

  ECOM100 #                      K0
  Workspace                    V503
  Success                      C100
  Error                        C101
  Error Code                  V2000
  Gateway Address       192.168.0.1
```

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

DS5/6	Used
HPP	N/A

ECOM100 Write IP Address will write the given IP Address to the ECOM100 module on a leading edge transition to the IBox. See also ECOM100 IP Setup (ECIPSUP) IBox 717 to setup ALL of the TCP/IP parameters in a single instruction - IP Address, Subnet Mask, and Gateway Address.



ECOM100 Write IP Address

ECWRIP IB-723

ECOM100 # K0

Workspace V400

Success C0

Error C0

Error Code V400

IP Address 0 . 0 . 0 . 0

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete. If there is an error, the Error Code parameter will report an ECOM100 error code (less than 100), or a PLC logic error (greater than 1000).

The IP Address is stored in Flash-ROM in the ECOM100 and the execution of this IBox will disable the ECOM100 module for at least a half second until it writes the Flash-ROM. Therefore, it is HIGHLY RECOMMENDED that you only execute this IBox ONCE on the second scan. Since it requires a LEADING edge to execute, use a NORMALLY CLOSED SP0 (STR NOT First Scan) to drive the power flow to the IBox.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

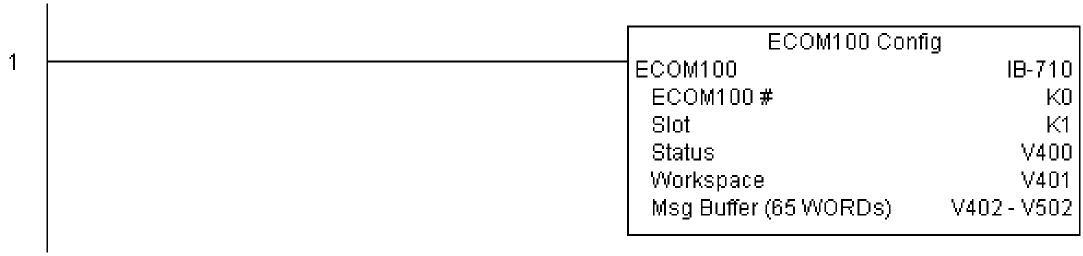
ECWRIP Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Error Code: specifies the location where the Error Code will be written
- IP Address: specifies the IP Address that will be written to the module

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error Code V	See DL405 V-memory map - Data Words
IP Address	0.0.0.1. to 255.255.255.254

ECWRIP Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



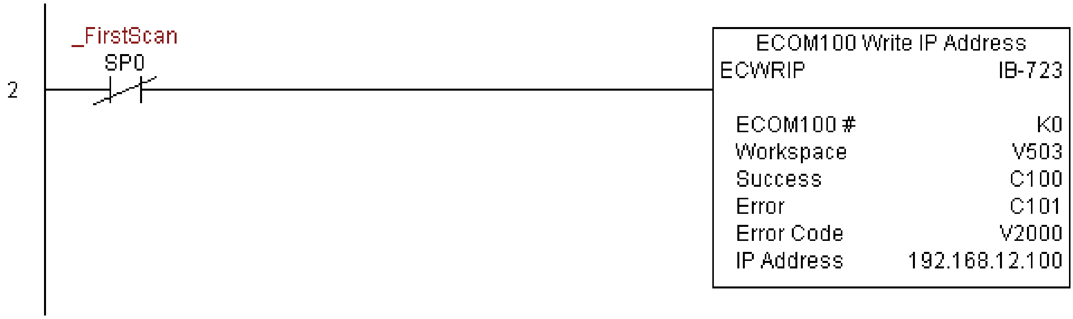
```
1                         ECOM100 Config
     ECOM100                                 IB-710
       ECOM100 #                                 K0
       Slot                                      K1
       Status                                  V400
       Workspace                               V401
       Msg Buffer (65 WORDs)            V402 - V502
```

Rung 2: On the 2nd scan, assign the IP Address of the ECOM100 to 192.168.12.100

The ECWRIP is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to write the IP Address will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON.

If successful, turn on C100. If there is a failure, turn on C101. If it fails, you can look at V2000 for the specific error code.

To configure all of the ECOM100 TCP/IP parameters in one IBox, see the ECOM100 IP Setup (ECIPSUP) IBox.



```
   _FirstScan
     SP0                  ECOM100 Write IP Address
2  --|/|--           ECWRIP                     IB-723

                       ECOM100 #                    K0
                       Workspace                  V503
                       Success                    C100
                       Error                      C101
                       Error Code                V2000
                       IP Address       192.168.12.100
```

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

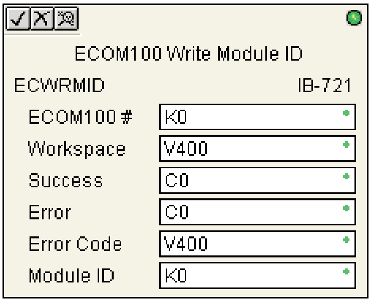
DS5/6	Used
HPP	N/A

ECOM100 Write Module ID will write the given Module ID on a leading edge transition to the IBox

If the Module ID is set in the hardware using the dipswitches, this IBox will fail and return error code 1005 (decimal).

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete. If there is an error, the Error Code parameter will report an ECOM100 error code (less than 100), or a PLC logic error (greater than 1000).



ECOM100 Write Module ID

ECWRMID IB-721

ECOM100 #	K0
Workspace	V400
Success	C0
Error	C0
Error Code	V400
Module ID	K0

The Module ID is stored in Flash-ROM in the ECOM100 and the execution of this IBox will disable the ECOM100 module for at least a half second until it writes the Flash-ROM. Therefore, it is HIGHLY RECOMMENDED that you only execute this IBox ONCE on the second scan. Since it requires a LEADING edge to execute, use a NORMALLY CLOSED SP0 (STR NOT First Scan) to drive the power flow to the IBox.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

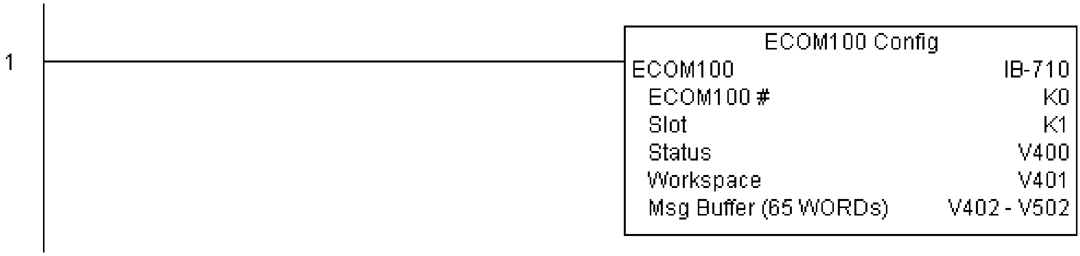
ECWRMID Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Error Code: specifies the location where the Error Code will be written
- Module ID: specifies the Module ID that will be written to the module

Parameter	DL405 Range
ECOM100# ....K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success ....X,Y,C,GX,GY,B	See DL405 V-memory map
Error ....X,Y,C,GX,GY,B	See DL405 V-memory map
Error Code V	See DL405 V-memory map - Data Words
Module ID	K0-65535

ECWRMID Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.

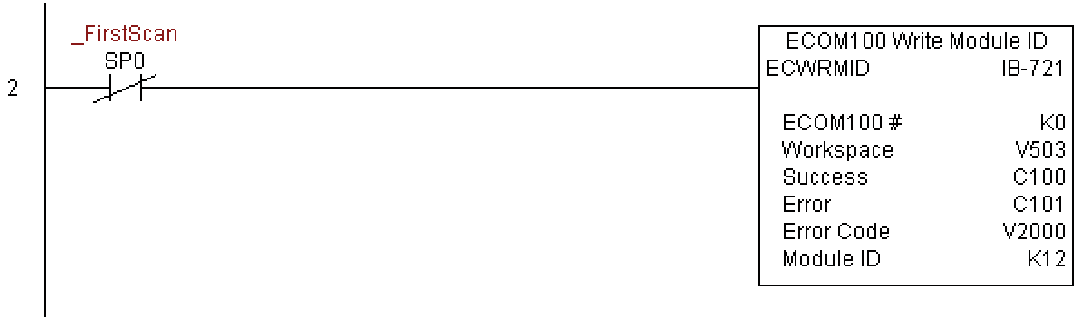


```
                 ECOM100 Config
ECOM100                              IB-710
  ECOM100 #                              K0
  Slot                                   K1
  Status                               V400
  Workspace                            V401
  Msg Buffer (65 WORDs)         V402 - V502
```

Rung 2: On the 2nd scan, set the Module ID of the ECOM100. Typically this is done using NetEdit, but this IBox allows you to configure the module ID of the ECOM100 using your ladder program.

The EWRMID is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to write the module ID will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON.

If successful, turn on C100. If there is a failure, turn on C101. If it fails, you can look at V2000 for the specific error code.



```
  ECOM100 Write Module ID
ECWRMID              IB-721

  ECOM100 #              K0
  Workspace            V503
  Success              C100
  Error                C101
  Error Code          V2000
  Module ID             K12
```

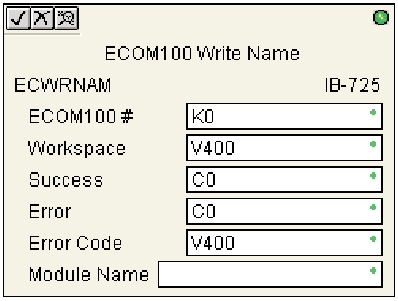
ECOM100 Write Name (ECWRNAM) (IB-725)

DS5/6	Used
HPP	N/A

ECOM100 Write Name will write the given Name to the ECOM100 module on a leading edge transition to the IBox. If you use a dollar sign ($) or double quote ("), use the PRINT/VPRINT escape sequence of TWO dollar signs ($$) for a single dollar sign or dollar sign-double quote ($") for a double quote character.

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete. If there is an error, the Error Code parameter will report an ECOM100 error code (less than 100), or a PLC logic error (greater than 1000).



ECOM100 Write Name

ECWRNAM IB-725

ECOM100 #	K0
Workspace	V400
Success	C0
Error	C0
Error Code	V400
Module Name	

The Name is stored in Flash-ROM in the ECOM100 and the execution of this IBox will disable the ECOM100 module for at least a half second until it writes the Flash-ROM. Therefore, it is HIGHLY RECOMMENDED that you only execute this IBox ONCE on the second scan. Since it requires a LEADING edge to execute, use a NORMALLY CLOSED SP0 (STR NOT First Scan) to drive the power flow to the IBox.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

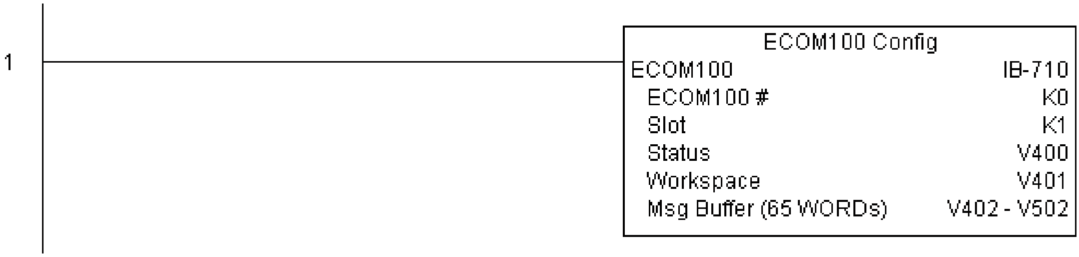
ECWRNAM Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Error Code: specifies the location where the Error Code will be written
- Module Name: specifies the Name that will be written to the module

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error Code V	See DL405 V-memory map - Data Words
Module Name	Text

ECWRNAM Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.

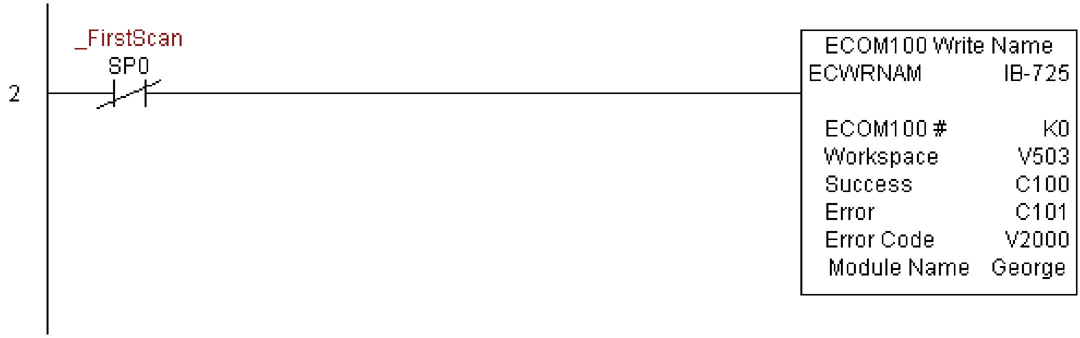


ECOM100 Config
ECOM100 IB-710
ECOM100 # K0
Slot K1
Status V400
Workspace V401
Msg Buffer (65 WORDs) V402 - V502

Rung 2: On the 2nd scan, set the Module Name of the ECOM100. Typically this is done using NetEdit, but this IBox allows you to configure the module name of the ECOM100 using your ladder program.

The EWRNAM is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to write the module name will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON.

If successful, turn on C100. If there is a failure, turn on C101. If it fails, you can look at V2000 for the specific error code.

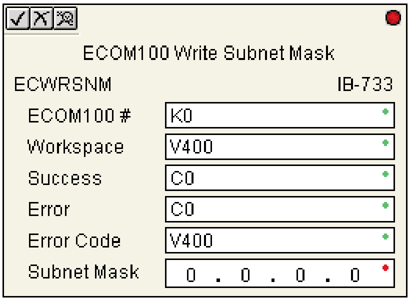


ECOM100 Write Name
ECWRNAM IB-725
ECOM100 # K0
Workspace V503
Success C100
Error C101
Error Code V2000
Module Name George

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

DS5/6	Used
HPP	N/A

ECOM100 Write Subnet Mask will write the given Subnet Mask to the ECOM100 module on a leading edge transition to the IBox. See also ECOM100 IP Setup (ECIPSUP) IBox 717 to setup ALL of the TCP/IP parameters in a single instruction - IP Address, Subnet Mask, and Gateway Address.



ECOM100 Write Subnet Mask

ECWRSNM IB-733

ECOM100 # K0

Workspace V400

Success C0

Error C0

Error Code V400

Subnet Mask 0 . 0 . 0 . 0

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Either the Success or Error bit parameter will turn on once the command is complete. If there is an error, the Error Code parameter will report an ECOM100 error code (less than 100), or a PLC logic error (greater than 1000).

The Subnet Mask is stored in Flash-ROM in the ECOM100 and the execution of this IBox will disable the ECOM100 module for at least a half second until it writes the Flash-ROM. Therefore, it is HIGHLY RECOMMENDED that you only execute this IBox ONCE on the second scan. Since it requires a LEADING edge to execute, use a NORMALLY CLOSED SP0 (STR NOT First Scan) to drive the power flow to the IBox.

In order for this ECOM100 IBox to function, you must turn ON dip switch 7 on the ECOM100 circuit board.

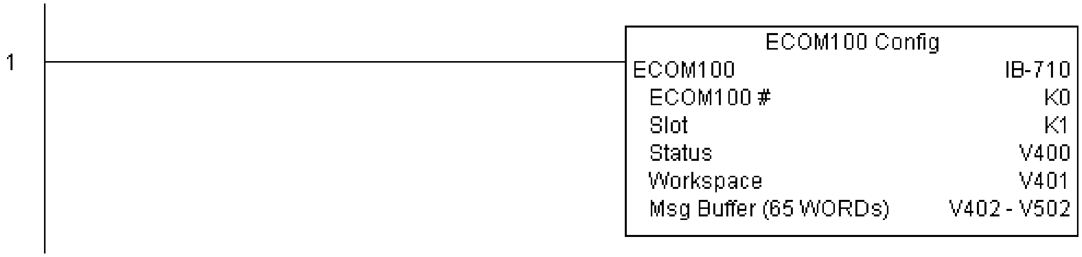
ECWRSNM Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed
- Error Code: specifies the location where the Error Code will be written
- Subnet Mask: specifies the Subnet Mask that will be written to the module

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error Code V	See DL405 V-memory map - Data Words
Subnet Mask	Masked IP Address

ECWRSNM Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



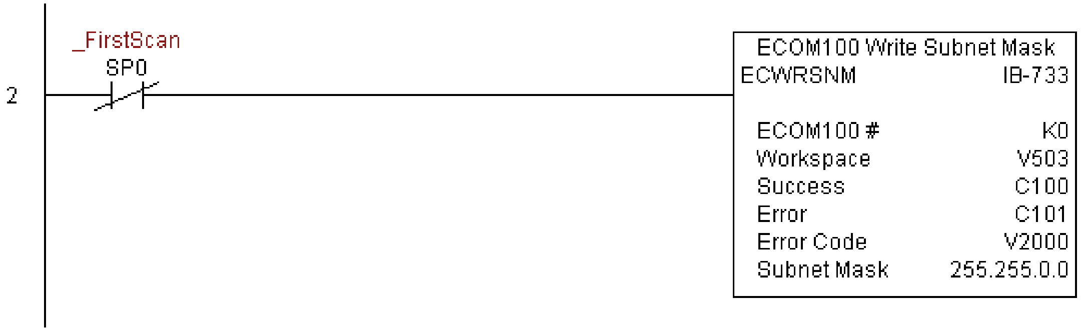
ECOM100 Config
ECOM100 IB-710
ECOM100 # K0
Slot K1
Status V400
Workspace V401
Msg Buffer (65 WORDs) V402 - V502

Rung 2: On the 2nd scan, assign the Subnet Mask of the ECOM100 to 255.255.0.0

The ECWRSNM is leading edge triggered, not power-flow driven (similar to a counter input leg). The command to write the Subnet Mask will be sent to the ECOM100 whenever the power flow into the IBox goes from OFF to ON.

If successful, turn on C100. If there is a failure, turn on C101. If it fails, you can look at V2000 for the specific error code.

To configure all of the ECOM100 TCP/IP parameters in one IBox, see the ECOM100 IP Setup (ECIPSUP) IBox.

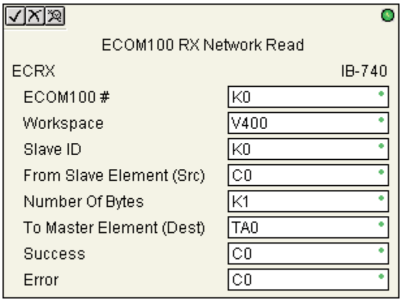


ECOM100 Write Subnet Mask
ECWRSNM IB-733
ECOM100 # K0
Workspace V503
Success C100
Error C101
Error Code V2000
Subnet Mask 255.255.0.0

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

DS5/6	Used
HPP	N/A

ECOM100 RX Network Read performs the RX instruction with built-in interlocking with all other ECOM100 RX (ECRX) and ECOM100 WX (ECWX) IBoxes in your program to simplify communications networking. It will perform the RX on the specified ECOM100#'s network, which corresponds to a specific unique ECOM100 Configuration (ECOM100) IBox at the top of your program.



ECOM100 RX Network Read

ECRX IB-740

ECOM100 #	K0
Workspace	V400
Slave ID	K0
From Slave Element (Src)	C0
Number Of Bytes	K1
To Master Element (Dest)	TA0
Success	C0
Error	C0

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Whenever this IBox has power, it will read element data from the specified slave into the given destination V memory buffer, giving other ECOM100 RX and ECOM100 WX IBoxes on that ECOM100# network a chance to execute.

For example, if you wish to read and write data continuously from 5 different slaves, you can have all of these ECRX and ECWX instructions in ONE RUNG driven by SP1 (Always On). They will execute round-robin style, automatically.

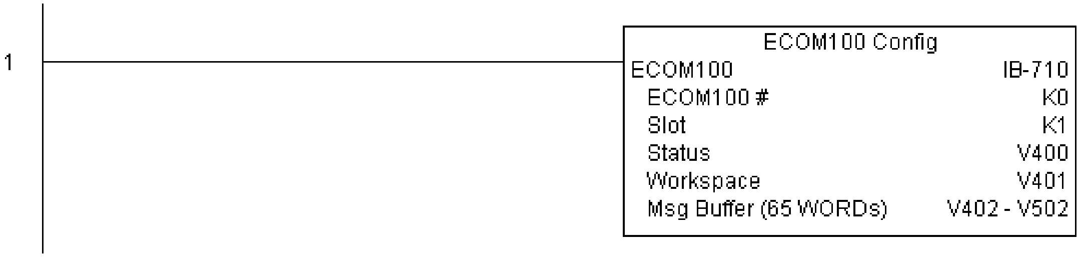
ECRX Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Slave ID: specifies the slave ECOM(100) PLC that will be targeted by the ECRX instruction
- From Slave Element (Src): specifies the slave address of the data to be read
- Number of Bytes: specifies the number of bytes to read from the slave ECOM(100) PLC
- To Master Element (Dest): specifies the location where the slave data will be placed in the master ECOM100 PLC
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Slave ID . . . . K	K0-90
From Slave Element (Src) X,Y,C,S,T,CT,GX,GY,V	See DL405 V-memory map
Number of Bytes . . . . K	K1-128
To Master Element (Dest) V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

ECRX Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



ECOM100 Config

ECOM100	IB-710
ECOM100 #	K0
Slot	K1
Status	V400
Workspace	V401
Msg Buffer (65 WORDs)	V402 - V502

(example continued on next page)

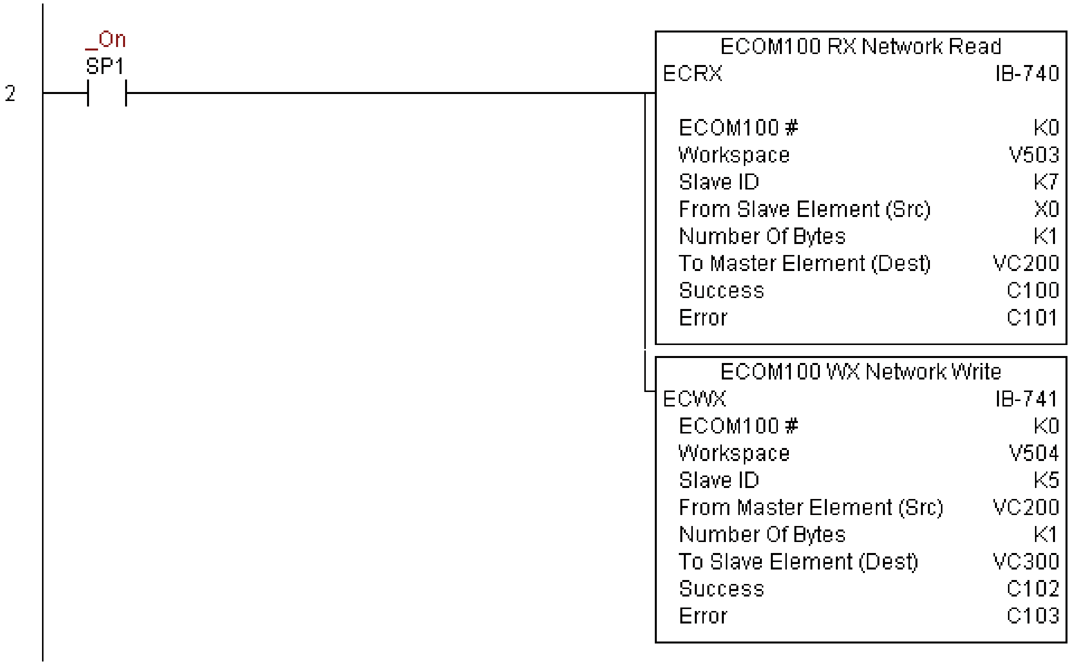
ECRX Example (con't)

Rung 2: Using ECOM100# K0, read X0-X7 from Slave K7 and write them to slave K5 as fast as possible. Store them in this local PLC in C200-C207, and write them to C300-C307 in slave K5.

Both the ECRX and ECWX work with the ECOM100 Config IBox to simplify all networking by handling all of the interlocks and proper resource sharing. They also provide very simplified error reporting. You no longer need to worry about any SP "busy bits" or "error bits", or what slot number a module is in, or have any counters or shift registers or any other interlocks for resource management.

In this example, SP1 (always ON) is driving both the ECRX and ECWX IBoxes in the same rung. On the scan that the Network Read completes, the Network Write will start that same scan. As soon as the Network Write completes, any pending operations below it in the program would get a turn. If there are no pending ECOM100 IBoxes below the ECWX, then the very next scan the ECRX would start its request again.

Using the ECRX and ECWX for all of your ECOM100 network reads and writes is the fastest the PLC can do networking. For local Serial Ports, DCM modules, or the original ECOM modules, use the NETCFG and NETRX/NETWX IBoxes.

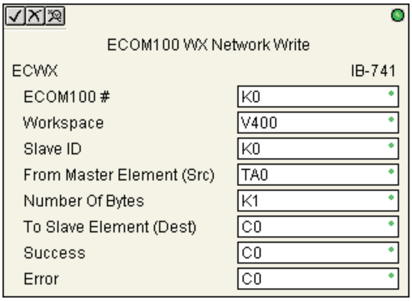


```
ECOM100 RX Network Read
ECRX                          IB-740

ECOM100 #                         K0
Workspace                       V503
Slave ID                          K7
From Slave Element (Src)          X0
Number Of Bytes                   K1
To Master Element (Dest)       VC200
Success                         C100
Error                           C101

ECOM100 WX Network Write
ECWX                          IB-741
ECOM100 #                         K0
Workspace                       V504
Slave ID                          K5
From Master Element (Src)      VC200
Number Of Bytes                   K1
To Slave Element (Dest)        VC300
Success                         C102
Error                           C103
```

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

DS5/6	Used
HPP	N/A

ECOM100 WX Network Write performs the WX instruction with built-in interlocking with all other ECOM100 RX (ECRX) and ECOM100 WX (ECWX) IBoxes in your program to simplify communications networking. It will perform the WX on the specified ECOM100#'s network, which corresponds to a specific unique ECOM100 Configuration (ECOM100) IBox at the top of your program.


ECOM100 WX Network Write
ECWX IB-741
ECOM100 # K0
Workspace V400
Slave ID K0
From Master Element (Src) TA0
Number Of Bytes K1
To Slave Element (Dest) C0
Success C0
Error C0

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Whenever this IBox has power, it will write data from the master's V memory buffer to the specified slave starting with the given slave element, giving other ECOM100 RX and ECOM100 WX IBoxes on that ECOM100# network a chance to execute.

For example, if you wish to read and write data continuously from 5 different slaves, you can have all of these ECRX and ECWX instructions in ONE RUNG driven by SP1 (Always On). They will execute round-robin style, automatically.

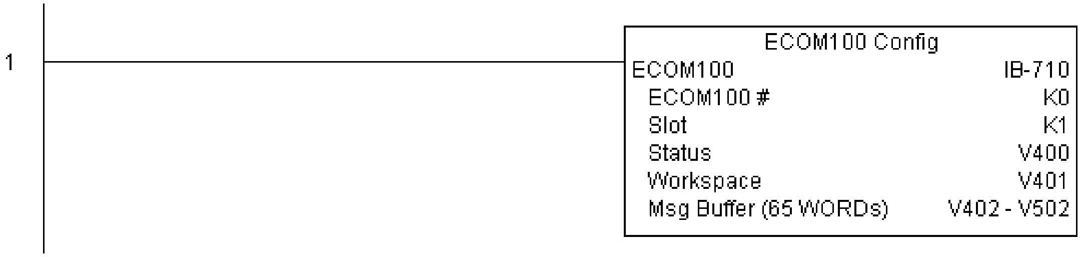
ECWX Parameters

- ECOM100#: this is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- Workspace: specifies a V-memory location that will be used by the instruction
- Slave ID: specifies the slave ECOM(100) PLC that will be targeted by the ECWX instruction
- From Master Element (Src): specifies the location in the master ECOM100 PLC where the data will be sourced from
- Number of Bytes: specifies the number of bytes to write to the slave ECOM(100) PLC
- To Slave Element (Dest): specifies the slave address the data will be written to
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Slave ID . . . . K	K0-90
From Master Element (Src) V	See DL405 V-memory map - Data Words
Number of Bytes . . . . K	K1-128
To Slave Element (Dest) . . X,Y,C,S,T,CT,GX,GY,V	See DL405 V-memory map
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

ECWX Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 1 as ECOM100# K0. All other ECxxxx IBoxes refer to this module # as K0. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V400 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module.V401 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V402-V502 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



```
                  ECOM100 Config
ECOM100                                IB-710
  ECOM100 #                                K0
  Slot                                     K1
  Status                                 V400
  Workspace                              V401
  Msg Buffer (65 WORDs)           V402 - V502
```

(example continued on next page)

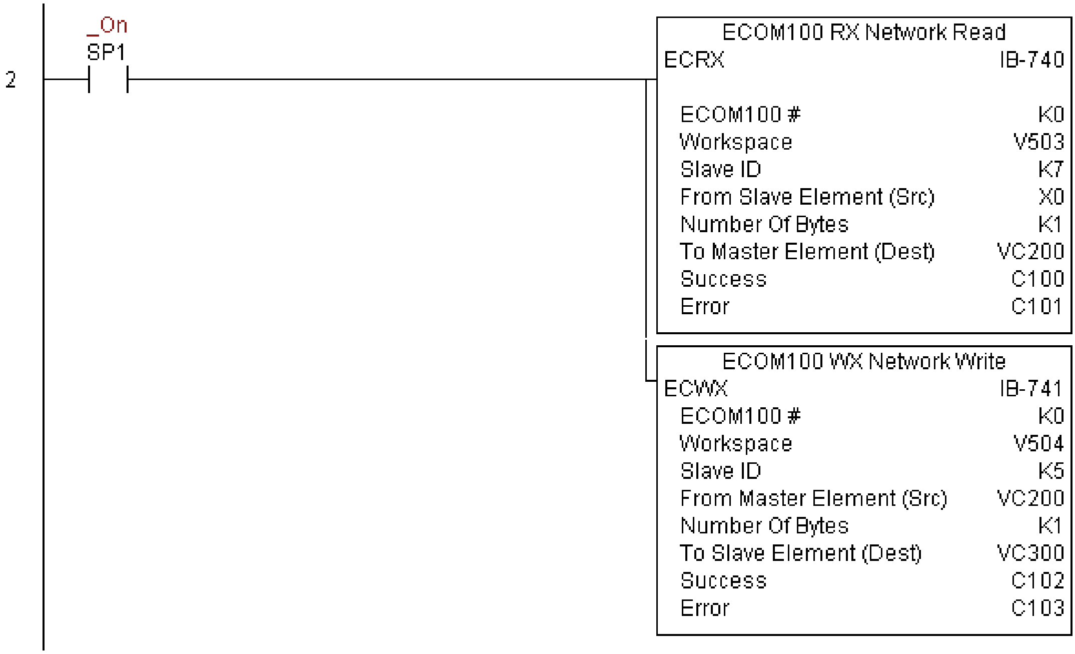
ECWX Example (con’t)

Rung 2: Using ECOM100# K0, read X0-X7 from Slave K7 and write them to slave K5 as fast as possible. Store them in this local PLC in C200-C207, and write them to C300-C307 in slave K5.

Both the ECRX and ECWX work with the ECOM100 Config IBox to simplify all networking by handling all of the interlocks and proper resource sharing. They also provide very simplified error reporting. You no longer need to worry about any SP "busy bits" or "error bits", or what slot number a module is in, or have any counters or shift registers or any other interlocks for resource management.

In this example, SP1 (always ON) is driving both the ECRX and ECWX IBoxes in the same rung. On the scan that the Network Read completes, the Network Write will start that same scan. As soon as the Network Write completes, any pending operations below it in the program would get a turn. If there are no pending ECOM100 IBoxes below the ECWX, then the very next scan the ECRX would start its request again.

Using the ECRX and ECWX for all of your ECOM100 network reads and writes is the fastest the PLC can do networking. For local Serial Ports, DCM modules, or the original ECOM modules, use the NETCFG and NETRX/NETWX IBoxes.



```
ECOM100 RX Network Read
ECRX                              IB-740

  ECOM100 #                           K0
  Workspace                         V503
  Slave ID                            K7
  From Slave Element (Src)            X0
  Number Of Bytes                     K1
  To Master Element (Dest)         VC200
  Success                           C100
  Error                             C101
```

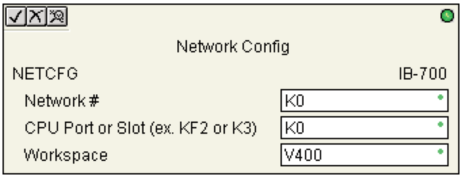
```
ECOM100 WX Network Write
ECWX                              IB-741
  ECOM100 #                           K0
  Workspace                         V504
  Slave ID                            K5
  From Master Element (Src)        VC200
  Number Of Bytes                     K1
  To Slave Element (Dest)          VC300
  Success                           C102
  Error                             C103
```

NETCFG Network Configuration (NETCFG) (IB-700)

DS5/6	Used
HPP	N/A

Network Config defines all the common information necessary for performing RX/WX Networking using the NETRX and NETWX IBox instructions via a local CPU serial port, DCM or ECOM module.

You must have the Network Config instruction at the top of your ladder/stage program with any other configuration IBoxes.



Network Config

NETCFG IB-700

Network # K0

CPU Port or Slot (ex. KF2 or K3) K0

Workspace V400

If you use more than one local serial port, DCM or ECOM in your PLC for RX/WX Networking, you must have a different Network Config instruction for EACH RX/WX network in your system that utilizes any NETRX/NETWX IBox instructions.

The Workspace parameter is an internal, private register used by the Network Config IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

The 2nd parameter "CPU Port or Slot" is the same value as in the high byte of the first LD instruction if you were coding the RX or WX rung yourself. This value is CPU port specific (check your PLC manual). Use KF1 or KF3 for the DL450 CPU ports 1 or 3. If using a DCM or ECOM module in the local base, use Kx, where x equals the slot where the module is installed. If using either module in an expansion base, use KXx, where X equals the expansion base number and x equals the slot in the expansion base where the module is installed.

NETCFG Parameters

- Network#: specifies a unique # for each ECOM(100) or DCM network to use
- CPU Port or Slot: specifies the CPU port number or slot number of DCM/ECOM(100) used

Parameter	DL405 Range
Network# . . . . K	K0-255
CPU Port or Slot . . . . K	K0-FF
Workspace V	See DL405 V-memory map - Data Words

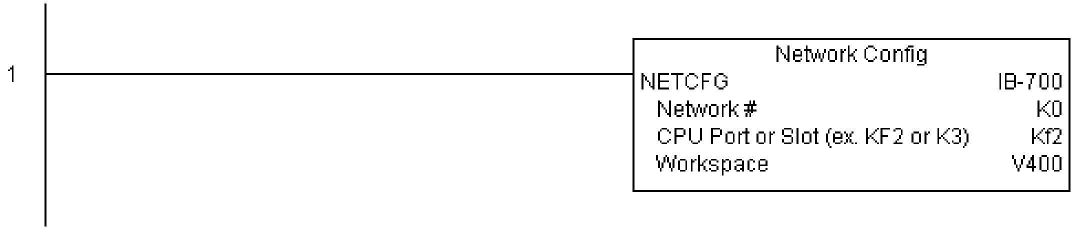
Workspace: specifies a V-memory location that will be used by the instruction

NETCFG Example

The Network Configuration IBox coordinates all of the interaction with other Network IBoxes (NETRX/NETWX). You must have a Network Configuration IBox for each serial port network, DCM module network, or original ECOM module network in your system. Configuration IBoxes must be at the top of your program and must execute every scan.

This IBox defines Network# K0 to be for the local CPU serial port #2 (KF2). For local CPU serial ports or DCM/ECOM modules, use the same value you would use in the most significant byte of the first LD instruction in a normal RX/WX rung to reference the port or module. Any NETRX or NETWX IBoxes that need to reference this specific network would enter K0 for their Network# parameter.

The Workspace register is used to maintain state information about the port or module, along with proper sharing and interlocking with the other NETRX and NETWX IBoxes in the program. This V-memory register must not be used anywhere else in the entire program.

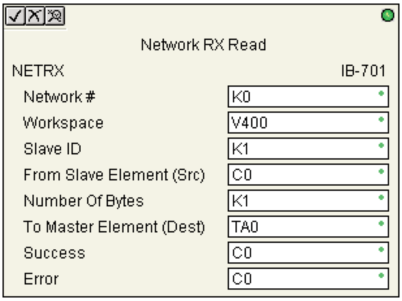


```
                 Network Config
NETCFG                                  IB-700
  Network #                                 K0
  CPU Port or Slot (ex. KF2 or K3)         Kf2
  Workspace                               V400
```

Network RX Read (NETRX) (IB-701)

DS5/6	Used
HPP	N/A

Network RX Read performs the RX instruction with built-in interlocking with all other Network RX (NETRX) and Network WX (NETWX) IBoxes in your program to simplify communications networking. It will perform the RX on the specified Network #, which corresponds to a specific unique Network Configuration (NETCFG) at the top of your program.



Network RX Read

NETRX	IB-701
Network #	K0
Workspace	V400
Slave ID	K1
From Slave Element (Src)	C0
Number Of Bytes	K1
To Master Element (Dest)	TA0
Success	C0
Error	C0

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Whenever this IBox has power, it will read element data from the specified slave into the given destination V memory buffer, giving other Network RX and Network WX IBoxes on that Network # a chance to execute.

For example, if you wish to read and write data continuously from 5 different slaves, you can have all of these NETRX and NETWX instructions in ONE RUNG driven by SP1 (Always On). They will execute round-robin style, automatically.

NETRX Parameters

- Network#: specifies the (CPU port's, DCM's, ECOM's) Network # defined by the NETCFG instruction
- Workspace: specifies a V-memory location that will be used by the instruction
- Slave ID: specifies the slave PLC that will be targeted by the NETRX instruction
- From Slave Element (Src): specifies the slave address of the data to be read
- Number of Bytes: specifies the number of bytes to read from the slave device
- To Master Element (Dest): specifies the location where the slave data will be placed in the master PLC
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed

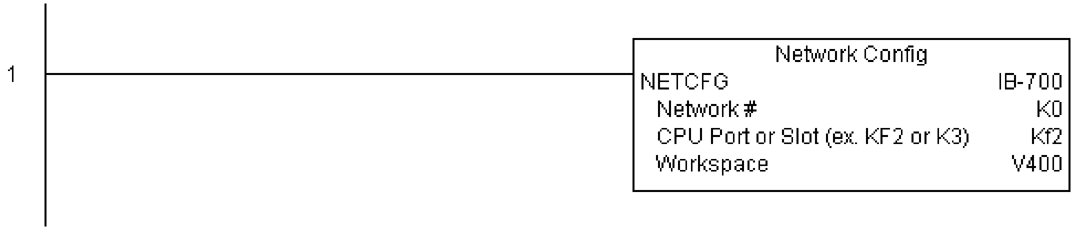
Parameter	DL405 Range
Network# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Slave ID . . . . K	K0-90
From Slave Element (Src) X,Y,C,S,T,CT,GX,GY,V	See DL405 V-memory map
Number of Bytes . . . . K	K1-128
To Master Element (Dest) V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

NETRX Example

Rung 1: The Network Configuration IBox coordinates all of the interaction with other Network IBoxes (NETRX/NETWX). You must have a Network Configuration IBox for each serial port network, DCM module network, or original ECOM module network in your system. Configuration IBoxes must be at the top of your program and must execute every scan.

This IBox defines Network# K0 to be for the local CPU serial port #2 (KF2). For local CPU serial ports or DCM/ECOM modules, use the same value you would use in the most significant byte of the first LD instruction in a normal RX/WX rung to reference the port or module. Any NETRX or NETWX IBoxes that need to reference this specific network would enter K0 for their Network# parameter.

The Workspace register is used to maintain state information about the port or module, along with proper sharing and interlocking with the other NETRX and NETWX IBoxes in the program. This V memory register must not be used anywhere else in the entire program.



```
1
                              Network Config
        NETCFG                                      IB-700
          Network #                                     K0
          CPU Port or Slot (ex. KF2 or K3)             Kf2
          Workspace                                   V400
```

(example continued on next page)

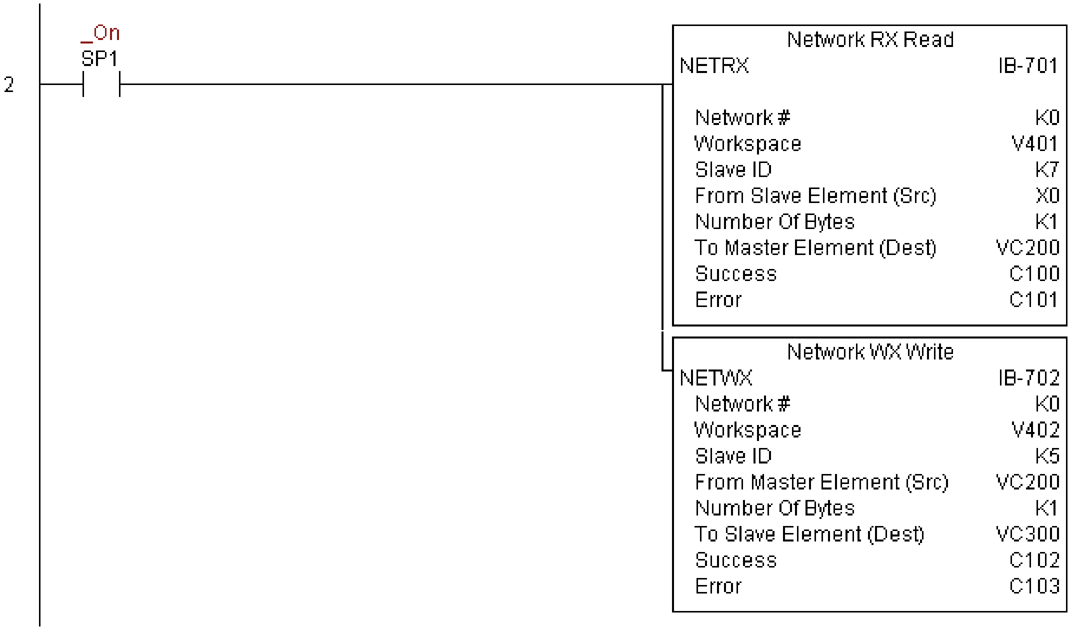
NETRX Example (con't)

Rung 2: Using Network# K0, read X0-X7 from Slave K7 and write them to slave K5 as fast as possible. Store them in this local PLC in C200-C207, and write them to C300-C307 in slave K5.

Both the NETRX and NETWX work with the Network Config IBox to simplify all networking by handling all of the interlocks and proper resource sharing. They also provide very simplified error reporting. You no longer need to worry about any SP "busy bits" or "error bits", or what port number or slot number a module is in, or have any counters or shift registers or any other interlocks for resource management.

In this example, SP1 (always ON) is driving both the NETRX and NETWX IBoxes in the same rung. On the scan that the Network Read completes, the Network Write will start that same scan. As soon as the Network Write completes, any pending operations below it in the program would get a turn. If there are no pending NETRX or NETWX IBoxes below this IBox, then the very next scan the NETRX would start its request again.

Using the NETRX and NETWX for all of your serial port, DCM, or original ECOM network reads and writes is the fastest the PLC can do networking. For ECOM100 modules, use the ECOM100 and ECRX/ECWX IBoxes.

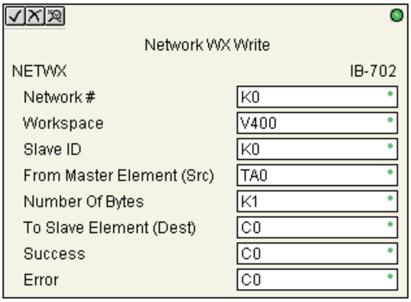


```
Network RX Read
NETRX                              IB-701

  Network #                            K0
  Workspace                          V401
  Slave ID                             K7
  From Slave Element (Src)             X0
  Number Of Bytes                      K1
  To Master Element (Dest)          VC200
  Success                            C100
  Error                              C101

Network WX Write
NETWX                              IB-702
  Network #                            K0
  Workspace                          V402
  Slave ID                             K5
  From Master Element (Src)         VC200
  Number Of Bytes                      K1
  To Slave Element (Dest)           VC300
  Success                            C102
  Error                              C103
```

Network WX Write (NETWX) (IB-702)

DS5/6	Used
HPP	N/A

Network WX Write performs the WX instruction with built-in interlocking with all other Network RX (NETRX) and Network WX (NETWX) IBoxes in your program to simplify communications networking. It will perform the WX on the specified Network #, which corresponds to a specific unique Network Configuration (NETCFG) at the top of your program.



Network WX Write

NETWX IB-702

Network #	K0
Workspace	V400
Slave ID	K0
From Master Element (Src)	TA0
Number Of Bytes	K1
To Slave Element (Dest)	C0
Success	C0
Error	C0

The Workspace parameter is an internal, private register used by this IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

Whenever this IBox has power, it will write data from the master's V memory buffer to the specified slave starting with the given slave element, giving other Network RX and Network WX IBoxes on that Network # a chance to execute.

For example, if you wish to read and write data continuously from 5 different slaves, you can have all of these NETRX and NETWX instructions in ONE RUNG driven by SP1 (Always On). They will execute round-robin style, automatically.

NETWX Parameters

- Network#: specifies the (CPU port's, DCM's, ECOM's) Network # defined by the NETCFG instruction
- Workspace: specifies a V-memory location that will be used by the instruction
- Slave ID: specifies the slave PLC that will be targeted by the NETWX instruction
- From Master Element (Src): specifies the location in the master PLC where the data will be sourced from
- Number of Bytes: specifies the number of bytes to write to the slave PLC
- To Slave Element (Dest): specifies the slave address the data will be written to
- Success: specifies a bit that will turn on once the request is completed successfully
- Error: specifies a bit that will turn on if the instruction is not successfully completed

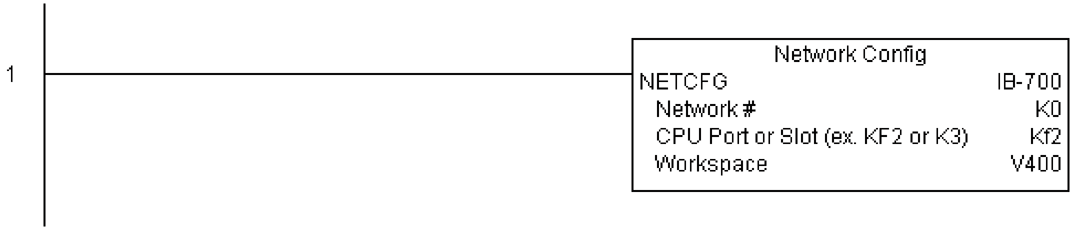
Parameter	DL405 Range
Network# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Slave ID . . . . K	K0-90
From Master Element (Src) V	See DL405 V-memory map - Data Words
Number of Bytes . . . . K	K1-128
To Slave Element (Dest) . . X,Y,C,S,T,CT,GX,GY,V	See DL405 V-memory map
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

NETWX Example

Rung 1: The Network Configuration IBox coordinates all of the interaction with other Network IBoxes (NETRX/NETWX). You must have a Network Configuration IBox for each serial port network, DCM module network, or original ECOM module network in your system. Configuration IBoxes must be at the top of your program and must execute every scan.

This IBox defines Network# K0 to be for the local CPU serial port #2 (KF2). For local CPU serial ports or DCM/ECOM modules, use the same value you would use in the most significant byte of the first LD instruction in a normal RX/WX rung to reference the port or module. Any NETRX or NETWX IBoxes that need to reference this specific network would enter K0 for their Network# parameter.

The Workspace register is used to maintain state information about the port or module, along with proper sharing and interlocking with the other NETRX and NETWX IBoxes in the program. This V memory register must not be used anywhere else in the entire program.



Network Config

NETCFG IB-700

Network # K0

CPU Port or Slot (ex. KF2 or K3) Kf2

Workspace V400

(example continued on next page)

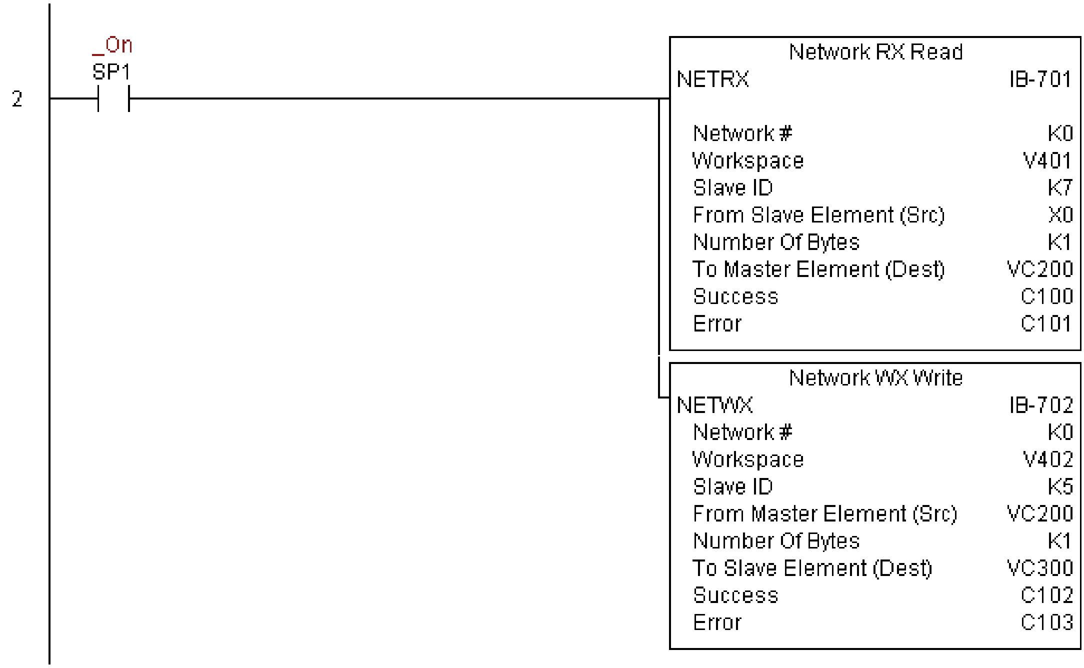
NETWX Example (con't)

Rung 2: Using Network# K0, read X0-X7 from Slave K7 and write them to slave K5 as fast as possible. Store them in this local PLC in C200-C207, and write them to C300-C307 in slave K5.

Both the NETRX and NETWX work with the Network Config IBox to simplify all networking by handling all of the interlocks and proper resource sharing. They also provide very simplified error reporting. You no longer need to worry about any SP "busy bits" or "error bits", or what port number or slot number a module is in, or have any counters or shift registers or any other interlocks for resource management.

In this example, SP1 (always ON) is driving both the NETRX and NETWX IBoxes in the same rung. On the scan that the Network Read completes, the Network Write will start that same scan. As soon as the Network Write completes, any pending operations below it in the program would get a turn. If there are no pending NETRX or NETWX IBoxes below this IBox, then the very next scan the NETRX would start its request again.

Using the NETRX and NETWX for all of your serial port, DCM, or original ECOM network reads and writes is the fastest the PLC can do networking. For ECOM100 modules, use the ECOM100 and ECRX/ECWX IBoxes.



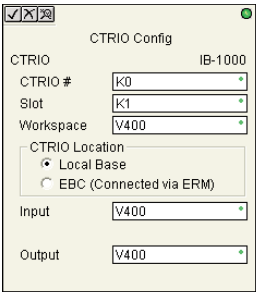
```
2   _On
    SP1
    
Network RX Read
NETRX                          IB-701

  Network #                        K0
  Workspace                      V401
  Slave ID                         K7
  From Slave Element (Src)         X0
  Number Of Bytes                  K1
  To Master Element (Dest)      VC200
  Success                        C100
  Error                          C101

Network WX Write
NETWX                          IB-702
  Network #                        K0
  Workspace                      V402
  Slave ID                         K5
  From Master Element (Src)     VC200
  Number Of Bytes                  K1
  To Slave Element (Dest)       VC300
  Success                        C102
  Error                          C103
```

CTRIO Configuration (CTRIO) (IB-1000)

DS5/6	Used
HPP	N/A

CTRIO Config defines all the common information for one specific CTRIO module which is used by the other CTRIO IBox instructions (for example, CTRLDPR - CTRIO Load Profile, CTREDRL - CTRIO Edit and Reload Preset Table, CTRRTLM - CTRIO Run to Limit Mode, ...).

The Input/Output parameters for this instruction can be copied directly from the CTRIO Workbench configuration for this CTRIO module. Since the behavior is slightly different when the CTRIO module is in an EBC Base via an ERM, you must specify whether the CTRIO module is in a local base or in an EBC base.



CTRIO Config

CTRIO IB-1000

CTRIO # K0

Slot K1

Workspace V400

CTRIO Location

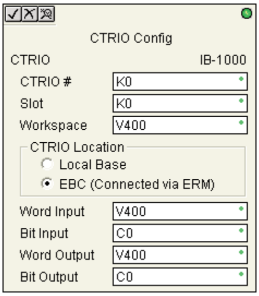
● Local Base

○ EBC (Connected via ERM)

Input V400

Output V400

CTRIO in Local Base



CTRIO Config

CTRIO IB-1000

CTRIO # K0

Slot K0

Workspace V400

CTRIO Location

○ Local Base

● EBC (Connected via ERM)

Word Input V400

Bit Input C0

Word Output V400

Bit Output C0

CTRIO in EBC Base

You must have the CTRIO Config IBox at the top of your ladder/stage program along with any other configuration IBoxes.

If you have more than one CTRIO in your PLC, you must have a different CTRIO Config IBox for EACH CTRIO module in your system that utilizes any CTRIO IBox instructions. Each CTRIO Config IBox must have a UNIQUE CTRIO# value. This is how the CTRIO IBoxes differentiate between the different CTRIO modules in your system.

The Workspace parameter is an internal, private register used by the CTRIO Config IBox and MUST BE UNIQUE in this one instruction and MUST NOT be used anywhere else in your program.

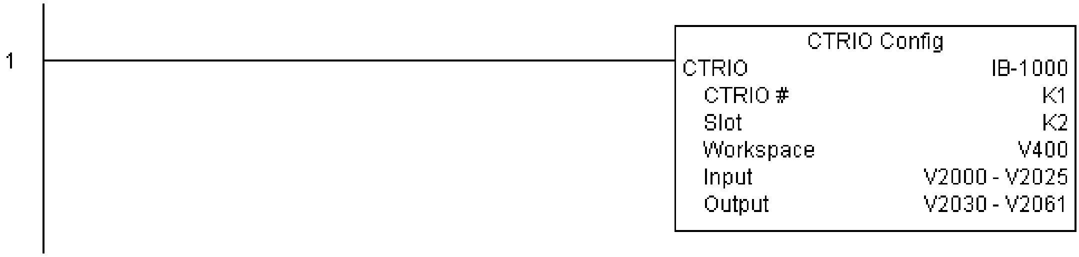
CTRIO Parameters

- CTRIO#: specifies a specific CTRIO module based on a user defined number
- Slot: (local base): specifies which PLC slot is occupied by the module (always K0 for EBC base)
- Workspace: specifies a V-memory location that will be used by the instruction
- CTRIO Location: specifies where the module is located (PLC local base or ERM to EBC base)
- Input (local base): This needs to be set to the same V-memory register as is specified in CTRIO Workbench as ‘Starting V address for inputs’ for this unique CTRIO.
- Output (local base): This needs to be set to the same V-memory register as is specified in CTRIO Workbench as ‘Starting V address for outputs’ for this unique CTRIO.
- Word Input (EBC base): The starting input V-memory address as defined by the I/O configuration in the ERM Workbench
- Bit Input (EBC base): The starting input Bit address as defined by the I/O configuration in the ERM Workbench
- Word Output (EBC base): The starting output V-memory address as defined by the I/O configuration in the ERM Workbench
- Bit Output (EBC base): The starting output Bit address as defined by the I/O configuration in the ERM Workbench

Parameter	DL205 Range
CTRIO# . . . . K	K0-255
Slot . . . . K	K0-7
Workspace V	See DL205 V-memory map - Data Words
Input (Word, Bit) V	See DL205 V-memory map - Data Words
Output (Word, Bit) V	See DL205 V-memory map - Data Words

CTRIO Example (local base)

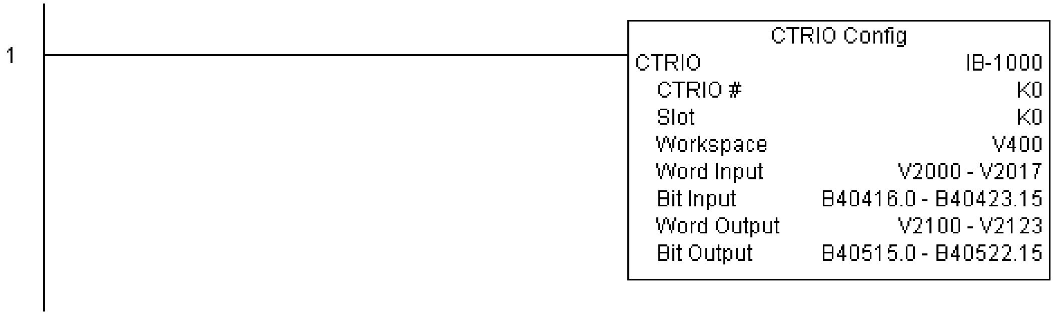
Rung 1: This sets up the CTRIO card in slot 2 of the local base. Each CTRIO in the system will need a separate CTRIO I-box before any CTRxxxx I-boxes can be used for them. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2030 through V2061 for its output data.



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

CTRIO Example (EBC base)

Overview: ERM Workbench must first be used to assign memory addresses to the I/O modules in the EBC base. Once the CTRIO module memory addresses are established using ERM Workbench, they are used in CTRIO Workbench and in a CTRIO IBox instruction to configure and define a specific CTRIO module. For this example, the CTRIO module uses V2000 - V2017 for its Word Input data and V40416.0 - V40423.15 for its Bit Input data. The module uses V2100 - V2123 for its Word Output data and V40515.0 - V40522.15 for its Bit Output data. The starting addresses, V2000 and V40416 (for inputs) and V2100 and V40515 (for outputs) are entered into CTRIO Workbench I/O Map to configure this specific CTRIO module. These starting addresses are the memory locations used in the CTRIO IBox instruction as the Word Input, Bit Input, Word Output and Bit Output addresses as shown below. For more information on this topic, refer to the CTRIO User Manual “Program Control” chapter.

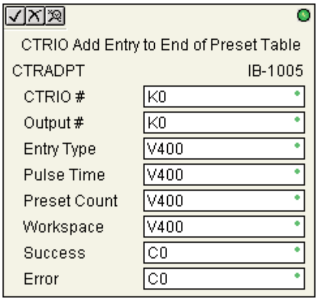


```
1   CTRIO Config
    CTRIO                          IB-1000
      CTRIO #                           K0
      Slot                              K0
      Workspace                       V400
      Word Input             V2000 - V2017
      Bit Input      B40416.0 - B40423.15
      Word Output            V2100 - V2123
      Bit Output     B40515.0 - B40522.15
```

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

DS5/6	Used
HPP	N/A

CTRIO Add Entry to End of Preset Table, on a leading edge transition to this IBox, will append an entry to the end of a memory based Preset Table on a specific CTRIO Output resource. This IBox will take more than 1 PLC scan to execute. Either the Success or Error bit will turn on when the command is complete. If the Error Bit is on, you can use the CTRIO Read Error Code (CTRRDER) IBox to get extended error information.



CTRIO Add Entry to End of Preset Table

CTRADPT IB-1005

CTRIO # K0

Output # K0

Entry Type V400

Pulse Time V400

Preset Count V400

Workspace V400

Success C0

Error C0

Entry Type:

K0: Set

K1: Reset

K2: Pulse On (uses Pulse Time)

K3: Pulse Off (uses Pulse Time)

K4: Toggle

K5: Reset Count

Note that the Pulse Time parameter is ignored by some Entry Types.

The Workspace register is for internal use by this IBox instruction and MUST NOT be used anywhere else in your program.

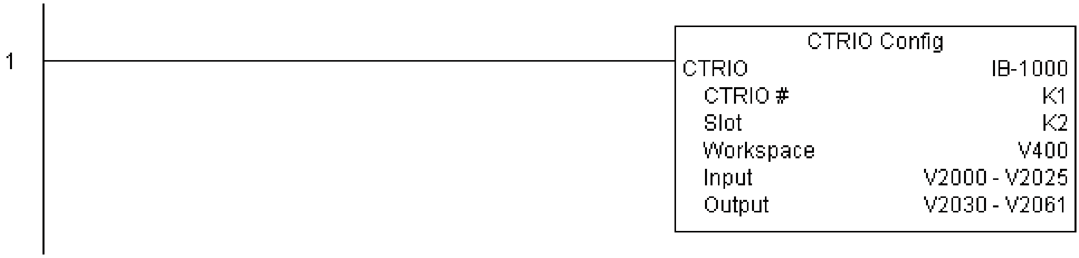
CTRADPT Parameters

- CTRIO#: specifies a specific CTRIO module based on a user defined number (see CTRIO Config)
- Output#: specifies a CTRIO output to be used by the instruction
- Entry Type: specifies the Entry Type to be added to the end of a Preset Table
- Pulse Time: specifies a pulse time for the Pulse On and Pulse Off Entry Types
- Preset Count: specifies an initial count value to begin at after Reset
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the instruction has successfully completed
- Error: specifies a bit that will turn on if the instruction does not complete successfully

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Entry Type . . . . V,K	K0-5; See DL405 V-memory map - Data Words
Pulse Time . . . . V,K	K0-65535; See DL405 V-memory map - Data Words
Preset Count . . . . V,K	K0-2147483647; See DL405 V-memory map
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

CTRADPT Example

Rung 1: This sets up the CTRIO card in slot 2 of the local base. Each CTRIO in the system will need a separate CTRIO I-box before any CTRxxxx I-boxes can be used for them. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2030 through V2061 for its output data.



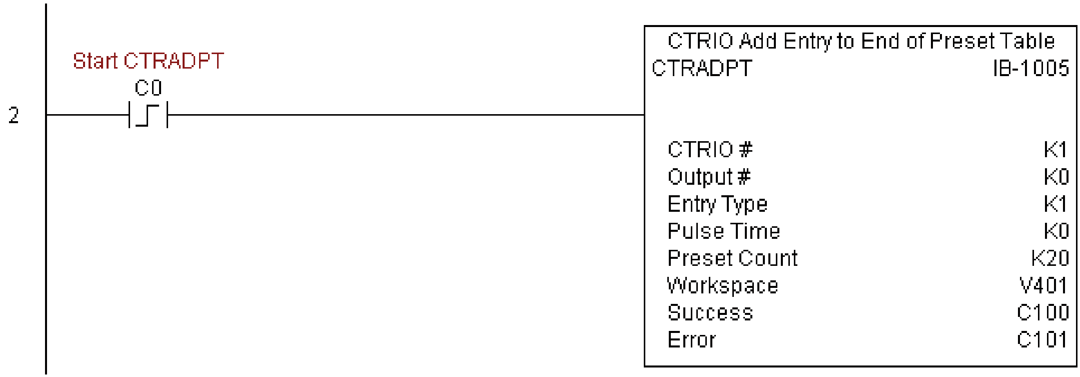
CTRIO Config

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

Rung 2: This rung is a sample method for enabling the CTRADPT command. A C-bit is used to allow the programmer to control the command from Data View for testing purposes.

Turning on C0 will cause the CTRADPT instruction to add a new preset to the preset table for output #0 on the CTRIO in slot 2. The new preset will be a command to RESET (entry type K1=reset), pulse time is left at zero as the reset type does not use this, and the count at which it will reset will be 20.

Operating procedure for this example code is to load the CTRADPT_ex1.cwb file to your CTRIO, then enter the code shown here, change to RUN mode, enable output #0 by turning on C2 in Data View, turn encoder on CTRIO to value above 10 and output #0 light will come on and stay on for all counts past 10. Now reset the counter with C1, enable C0 to execute CTRADPT command to add a reset for output #0 at a count of 20, turn on C2 to enable output #0, then turn encoder to value of 10+ (output #0 should turn on) and then continue on to count of 20+ (output #0 should turn off).



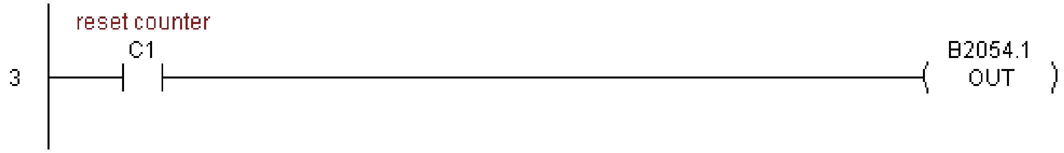
CTRIO Add Entry to End of Preset Table

CTRADPT	IB-1005
CTRIO #	K1
Output #	K0
Entry Type	K1
Pulse Time	K0
Preset Count	K20
Workspace	V401
Success	C100
Error	C101

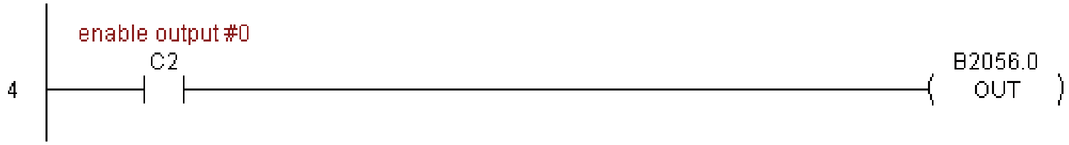
(example continued on next page)

CTRADPT Example (con't)

Rung 3: This rung allows the programmer to reset the counter from the ladder logic.



Rung 4: This rung allows the operator to enable output #0 from the ladder code.

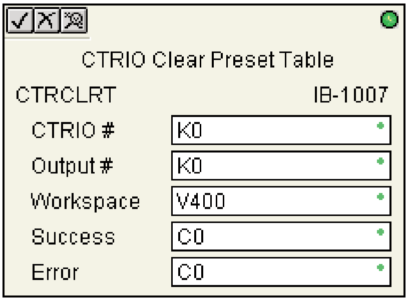


CTRIO Clear Preset Table (CTRCLRT) (IB-1007)

DS5/6	Used
HPP	N/A

CTRIO Clear Preset Table will clear the RAM based Preset Table on a leading edge transition to this IBox. This IBox will take more than 1 PLC scan to execute. Either the Success or Error bit will turn on when the command is complete. If the Error Bit is on, you can use the CTRIO Read Error Code (CTRRDER) IBox to get extended error information.

The Workspace register is for internal use by this IBox instruction and MUST NOT be used anywhere else in your program.



CTRIO Clear Preset Table

CTRCLRT IB-1007

CTRIO # K0

Output # K0

Workspace V400

Success C0

Error C0

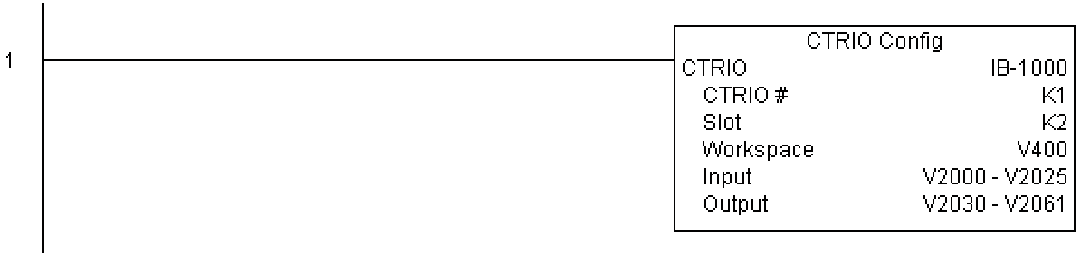
CTRCLRT Parameters

- CTRIO#: specifies a specific CTRIO module based on a user defined number (see CTRIO Config)
- Output#: specifies a CTRIO output to be used by the instruction
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the instruction has successfully completed
- Error: specifies a bit that will turn on if the instruction does not complete successfully

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

CTRCLRT Example

Rung 1: This sets up the CTRIO card in slot 2 of the local base. Each CTRIO in the system will need a separate CTRIO I-box before any CTRxxxx I-boxes can be used for them. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2030 through V2061 for its output data.



CTRIO Config

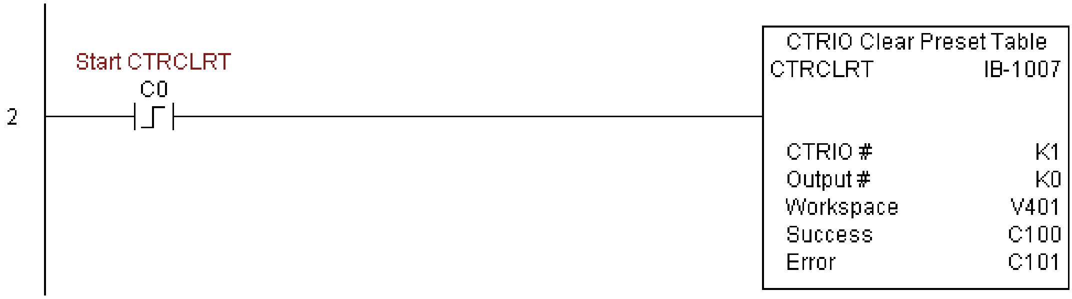
CTRIO IB-1000

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

Rung 2: This rung is a sample method for enabling the CTRCLRT command. A C-bit is used to allow the programmer to control the command from Data View for testing purposes.

Turning on C0 will cause the CTRCLRT instruction to clear the preset table for output #0 on the CTRIO in slot 2.

Operating procedure for this example code is to load the CTRCLRT_ex1.cwb file to your CTRIO, then enter the code shown here, change to RUN mode, enable output #0 by turning on C2 in Data View, turn encoder on CTRIO to value above 10 and output #0 light will come on and stay on until a count of 20 is reached, where it will turn off. Now reset the counter with C1, enable C0 to execute CTRCLRT command to clear the preset table, turn on C2 to enable output #0, then turn encoder to value of 10+ (output #0 should NOT turn on).



CTRIO Clear Preset Table

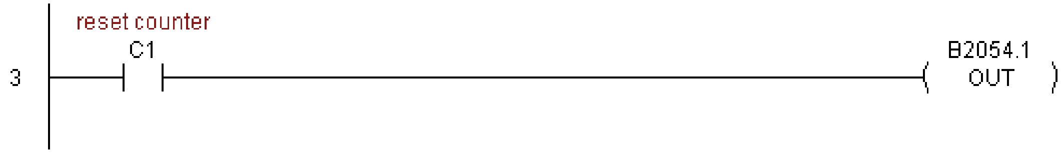
CTRCLRT IB-1007

- CTRIO # K1
- Output # K0
- Workspace V401
- Success C100
- Error C101

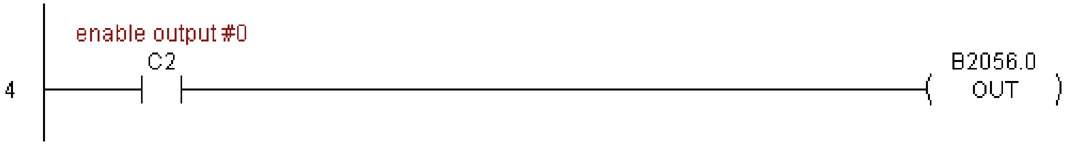
(example continued on next page)

CTRCLRT Example (con't)

Rung 3: This rung allows the programmer to reset the counter from the ladder logic.



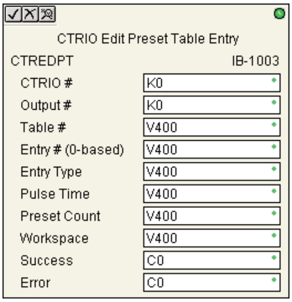
Rung 4: This rung allows the operator to enable output #0 from the ladder code.



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

DS5/6	Used
HPP	N/A

CTRIO Edit Preset Table Entry, on a leading edge transition to this IBox, will edit a single entry in a Preset Table on a specific CTRIO Output resource. This IBox is good if you are editing more than one entry in a file at a time. If you wish to do just one edit and then reload the table immediately, see the CTRIO Edit and Reload Preset Table Entry (CTREDRL) IBox. This IBox will take more than 1 PLC scan to execute. Either the Success or Error bit will turn on when the command is complete. If the Error Bit is on, you can use the CTRIO Read Error Code (CTRRDER) IBox to get extended error information.



CTRIO Edit Preset Table Entry

CTREDPT IB-1003

CTRIO #	K0
Output #	K0
Table #	V400
Entry # (0-based)	V400
Entry Type	V400
Pulse Time	V400
Preset Count	V400
Workspace	V400
Success	C0
Error	C0

Entry Type:

K0: Set

K1: Reset

K2: Pulse On (uses Pulse Time)

K3: Pulse Off (uses Pulse Time)

K4: Toggle

K5: Reset Count

Note that the Pulse Time parameter is ignored by some Entry Types.

The Workspace register is for internal use by this IBox instruction and MUST NOT be used anywhere else in your program.

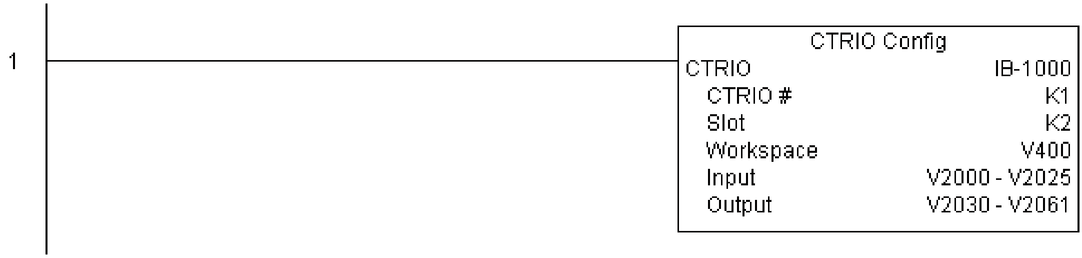
CTREDPT Parameters

- CTRIO#: specifies a specific CTRIO module based on a user defined number (see CTRIO Config Ibox)
- Output#: specifies a CTRIO output to be used by the instruction
- Table#: specifies the Table number of which an Entry is to be edited
- Entry#: specifies the Entry location in the Preset Table to be edited
- Entry Type: specifies the Entry Type to add during the edit
- Pulse Time: specifies a pulse time for the Pulse On and Pulse Off Entry Types
- Preset Count: specifies an initial count value to begin at after Reset
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the instruction has successfully completed
- Error: specifies a bit that will turn on if the instruction does not complete successfully

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Table# . . . . V,K	K0-255; See DL405 V-memory map - Data Words
Entry# . . . . V,K	K0-255; See DL405 V-memory map - Data Words
Entry Type . . . . V,K	K0-5; See DL405 V-memory map - Data Words
Pulse Time . . . . V,K	K0-65535; See DL405 V-memory map - Data Words
Preset Count . . . . V,K	K0-2147483647; See DL405 V-memory map
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

CTREDPT Example

Rung 1: This sets up the CTRIO card in slot 2 of the local base. Each CTRIO in the system will need a separate CTRIO I-box before any CTRxxxx I-boxes can be used for them. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2030 through V2061 for its output data.



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

(example continued on next page)

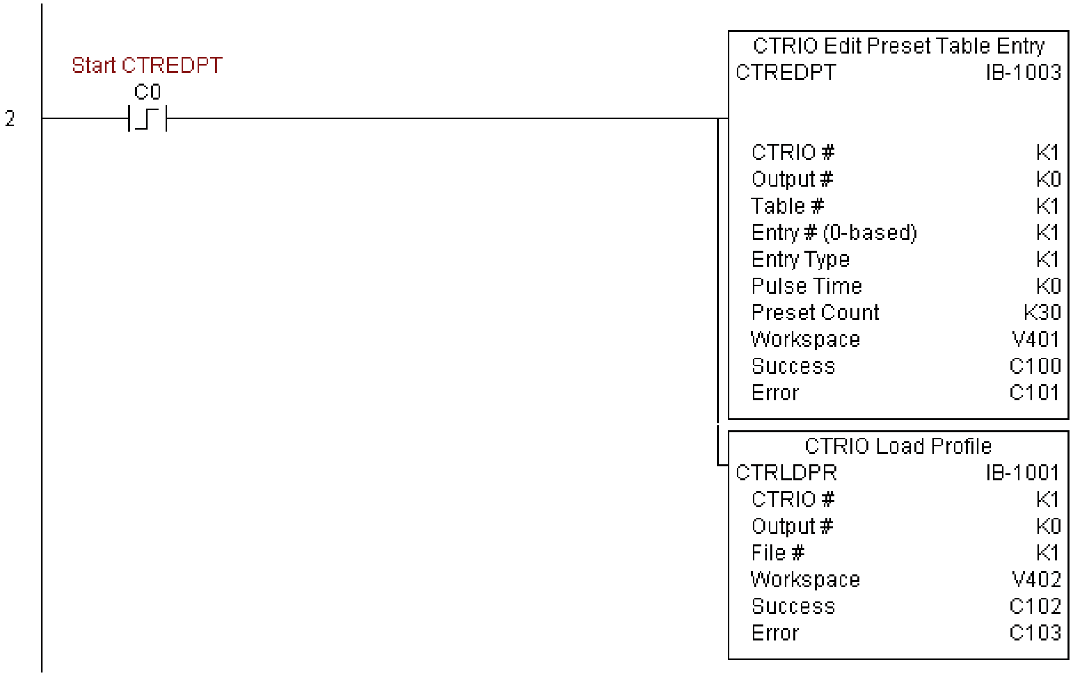
CTREDPT Example (con’t)

Rung 2: This rung is a sample method for enabling the CTREDPT command. A C-bit is used to allow the programmer to control the command from Data View for testing purposes.

Turning on C0 will cause the CTREDPT instruction to change the second preset from a reset at a count of 20 to a reset at a count of 30 for output #0 on the CTRIO in slot 2.

Operating procedure for this example code is to load the CTREDPT_ex1.cwb file to your CTRIO, then enter the code shown here, change to RUN mode, enable output #0 by turning on C2 in Data View, turn encoder on CTRIO to value above 10 and output #0 light will come on and stay on until a count of 20 is reached, where it will turn off. Now reset the counter with C1, enable C0 to execute CTREDPT command to change the second preset, turn on C2 to enable output #0, then turn encoder to value of 10+ (output #0 should turn on) and then continue past a count of 30 (output #0 should turn off).

Note that we must also reload the profile after changing the preset(s), this is why the CTRLDPR command follows the CTREDPT command in this example.

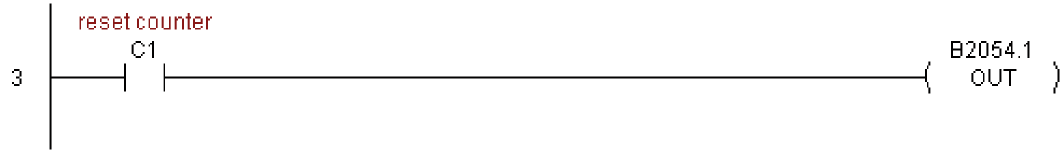


```
CTRIO Edit Preset Table Entry
CTREDPT                 IB-1003

  CTRIO #                    K1
  Output #                   K0
  Table #                    K1
  Entry # (0-based)          K1
  Entry Type                 K1
  Pulse Time                 K0
  Preset Count              K30
  Workspace                V401
  Success                  C100
  Error                    C101

       CTRIO Load Profile
CTRLDPR                 IB-1001
  CTRIO #                    K1
  Output #                   K0
  File #                     K1
  Workspace                V402
  Success                  C102
  Error                    C103
```

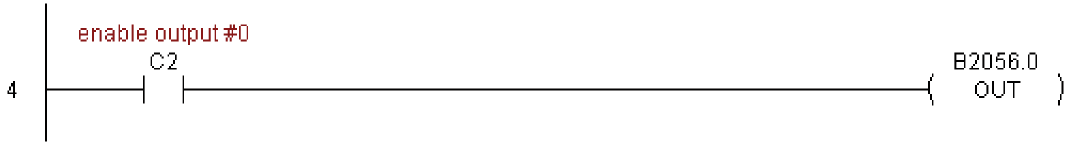
(example continued on next page)

CTREDPT Example (con't)

Rung 3: This rung allows the programmer to reset the counter from the ladder logic.



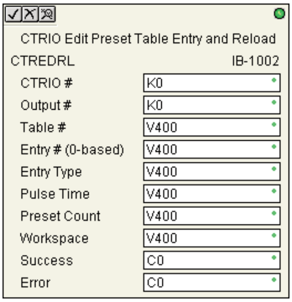
Rung 4: This rung allows the operator to enable output #0 from the ladder code.



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

DS5/6	Used
HPP	N/A

CTRIO Edit Preset Table Entry and Reload, on a leading edge transition to this IBox, will perform this dual operation to a CTRIO Output resource in one CTRIO command. This IBox will take more than 1 PLC scan to execute. Either the Success or Error bit will turn on when the command is complete. If the Error Bit is on, you can use the CTRIO Read Error Code (CTRRDER) IBox to get extended error information.



CTRIO Edit Preset Table Entry and Reload

CTREDRL IB-1002

CTRIO #	K0
Output #	K0
Table #	V400
Entry # (0-based)	V400
Entry Type	V400
Pulse Time	V400
Preset Count	V400
Workspace	V400
Success	C0
Error	C0

Entry Type:

K0: Set

K1: Reset

K2: Pulse On (uses Pulse Time)

K3: Pulse Off (uses Pulse Time)

K4: Toggle

K5: Reset Count

Note that the Pulse Time parameter is ignored by some Entry Types.

The Workspace register is for internal use by this IBox instruction and MUST NOT be used anywhere else in your program.

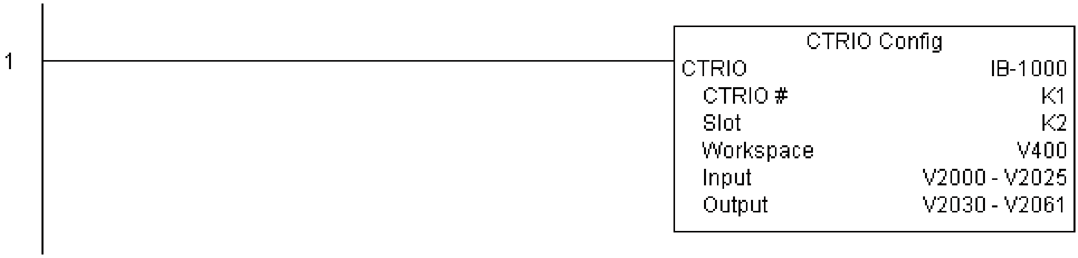
CTREDRL Parameters

- CTRIO#: specifies a specific CTRIO module based on a user defined number (see CTRIO Config Ibox)
- Output#: specifies a CTRIO output to be used by the instruction
- Table#: specifies the Table number of which an Entry is to be edited
- Entry#: specifies the Entry location in the Preset Table to be edited
- Entry Type: specifies the Entry Type to add during the edit
- Pulse Time: specifies a pulse time for the Pulse On and Pulse Off Entry Types
- Preset Count: specifies an initial count value to begin at after Reset
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the instruction has successfully completed
- Error: specifies a bit that will turn on if the instruction does not complete successfully

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Table# . . . . V,K	K0-255; See DL405 V-memory map - Data Words
Entry# . . . . V,K	K0-255; See DL405 V-memory map - Data Words
Entry Type . . . . V,K	K0-5; See DL405 V-memory map - Data Words
Pulse Time . . . . V,K	K0-65535; See DL405 V-memory map - Data Words
Preset Count . . . . V,K	K0-2147483647; See DL405 V-memory map
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

CTREDRL Example

Rung 1: This sets up the CTRIO card in slot 2 of the local base. Each CTRIO in the system will need a separate CTRIO I-box before any CTRxxxx I-boxes can be used for them. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2030 through V2061 for its output data.



(example continued on next page)

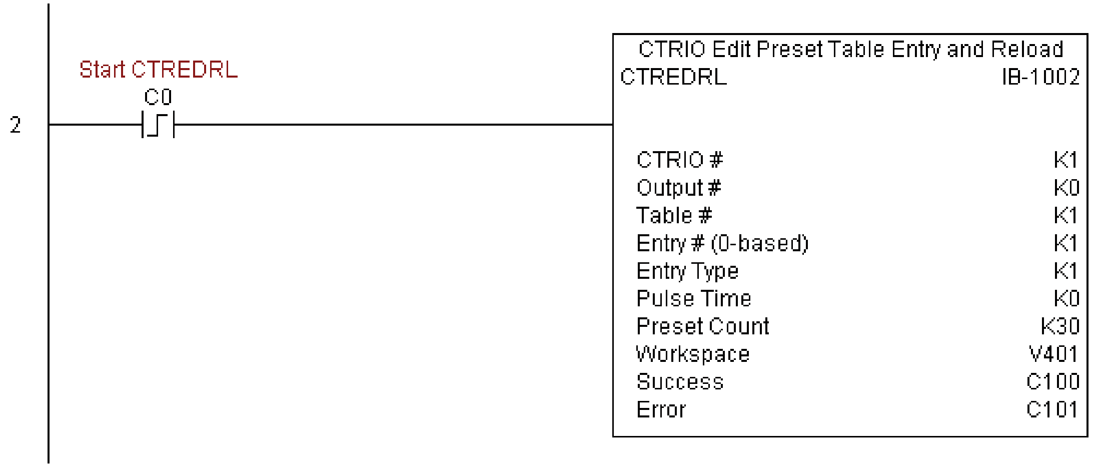
CTREDRL Example (con't)

Rung 2: This rung is a sample method for enabling the CTREDRL command. A C-bit is used to allow the programmer to control the command from Data View for testing purposes.

Turning on C0 will cause the CTREDRL instruction to change the second preset in file 1 from a reset at a value of 20 to a reset at a value of 30.

Operating procedure for this example code is to load the CTREDRL_ex1.cwb file to your CTRIO, then enter the code shown here, change to RUN mode, enable output #0 by turning on C2 in Data View, turn encoder on CTRIO to value above 10 and output #0 light will come on, continue to a count above 20 and the output #0 light will turn off. Now reset the counter with C1, enable C0 to execute CTREDRL command to change the second preset count value to 30, then turn encoder to value of 10+ (output #0 should turn on) and continue on to a value of 30+ and the output #0 light will turn off.

Note that it is not necessary to reload this file separately, however, the command can only change one value at a time.



2

Start CTREDRL
C0

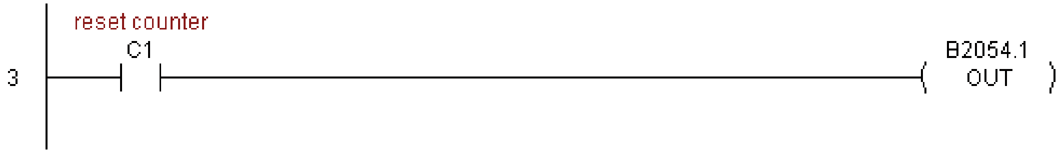
CTRIO Edit Preset Table Entry and Reload
CTREDRL IB-1002

CTRIO #	K1
Output #	K0
Table #	K1
Entry # (0-based)	K1
Entry Type	K1
Pulse Time	K0
Preset Count	K30
Workspace	V401
Success	C100
Error	C101

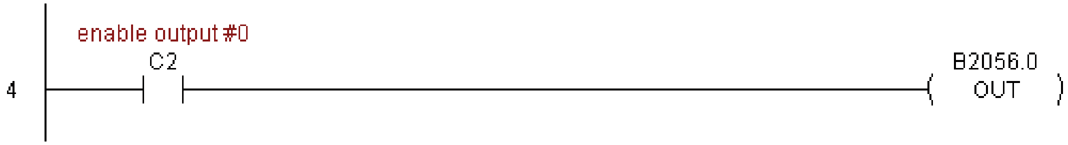
(example continued on next page)

CTREDRL Example (con't)

Rung 3: This rung allows the programmer to reset the counter from the ladder logic.



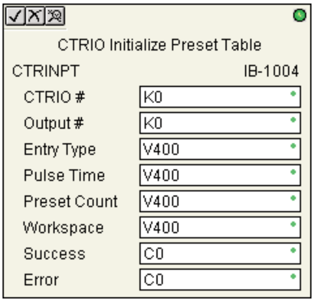
Rung 4: This rung allows the operator to enable output #0 from the ladder code.



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

DS5/6	Used
HPP	N/A

CTRIO Initialize Preset Table, on a leading edge transition to this IBox, will create a single entry Preset Table in memory but not as a file, on a specific CTRIO Output resource. This IBox will take more than 1 PLC scan to execute. Either the Success or Error bit will turn on when the command is complete. If the Error Bit is on, you can use the CTRIO Read Error Code (CTRRDER) IBox to get extended error information.



CTRIO Initialize Preset Table

CTRINPT IB-1004

CTRIO #	K0
Output #	K0
Entry Type	V400
Pulse Time	V400
Preset Count	V400
Workspace	V400
Success	C0
Error	C0

Entry Type:

K0: Set

K1: Reset

K2: Pulse On (uses Pulse Time)

K3: Pulse Off (uses Pulse Time)

K4: Toggle

K5: Reset Count

Note that the Pulse Time parameter is ignored by some Entry Types.

The Workspace register is for internal use by this IBox instruction and MUST NOT be used anywhere else in your program.

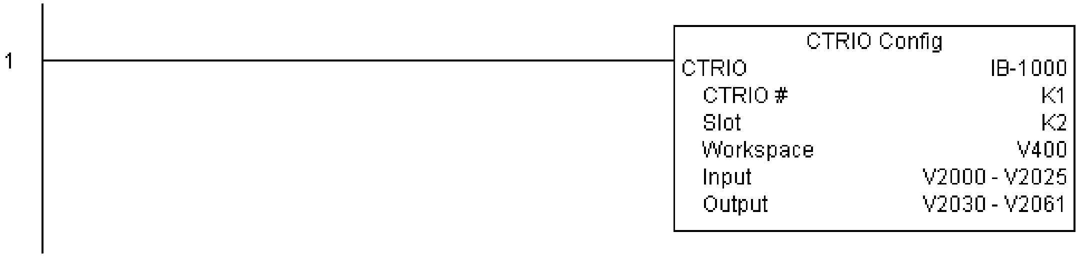
CTRINPT Parameters

- CTRIO#: specifies a specific CTRIO module based on a user defined number (see CTRIO Config Ibox)
- Output#: specifies a CTRIO output to be used by the instruction
- Entry Type: specifies the Entry Type to add during the edit
- Pulse Time: specifies a pulse time for the Pulse On and Pulse Off Entry Types
- Preset Count: specifies an initial count value to begin at after Reset
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the instruction has successfully completed
- Error: specifies a bit that will turn on if the instruction does not complete successfully

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Entry Type . . . . V,K	K0-5; See DL405 V-memory map - Data Words
Pulse Time . . . . V,K	K0-65535; See DL405 V-memory map - Data Words
Preset Count . . . . V,K	K0-2147483647; See DL405 V-memory map
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

CTRINPT Example

Rung 1: This sets up the CTRIO card in slot 2 of the local base. Each CTRIO in the system will need a separate CTRIO I-box before any CTRxxxx I-boxes can be used for them. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2030 through V2061 for its output data.



```
1
                                        CTRIO Config
                              CTRIO                       IB-1000
                                CTRIO #                        K1
                                Slot                           K2
                                Workspace                    V400
                                Input               V2000 - V2025
                                Output              V2030 - V2061
```

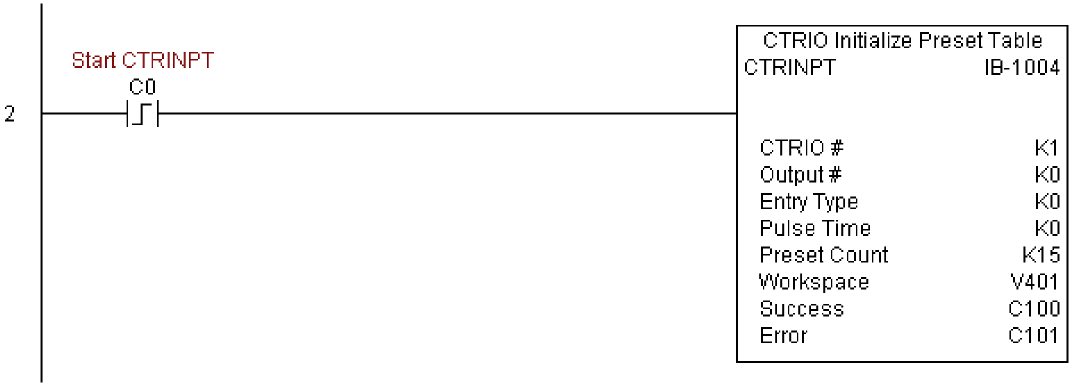
(example continued on next page)

CTRINPT Example (con't)

Rung 2: This rung is a sample method for enabling the CTRINPT command. A C-bit is used to allow the programmer to control the command from Data View for testing purposes.

Turning on C0 will cause the CTRINPT instruction to create a single entry preset table, but not as a file, and use it for the output #0. In this case the single preset will be a set at a count of 15 for output #0.

Operating procedure for this example code is to load the CTRINPT_ex1.cwb file to your CTRIO, then enter the code shown here, change to RUN mode, enable output #0 by turning on C2 in Data View, turn encoder on CTRIO to value above 15 and output #0 light will not come on. Now reset the counter with C1, enable C0 to execute CTRINPT command to create a single preset table with a preset to set output#0 at a count of 15, then turn encoder to value of 15+ (output #0 should turn on).



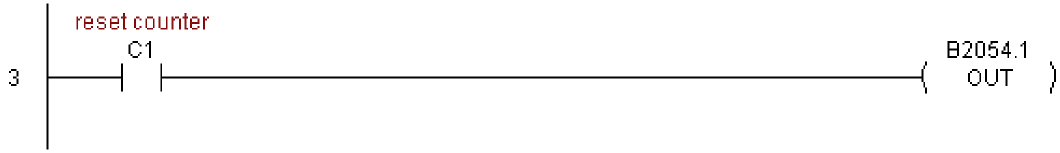
CTRIO Initialize Preset Table
CTRINPT IB-1004

CTRIO #	K1
Output #	K0
Entry Type	K0
Pulse Time	K0
Preset Count	K15
Workspace	V401
Success	C100
Error	C101

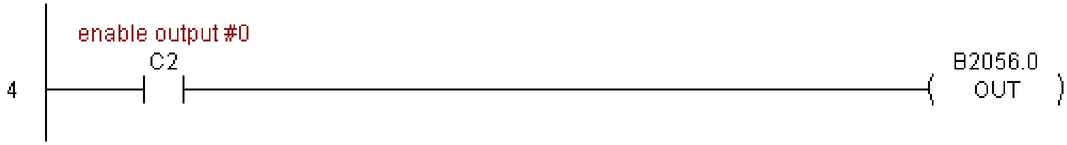
(example continued on next page)

CTRINPT Example (con't)

Rung 3: This rung allows the programmer to reset the counter from the ladder logic.



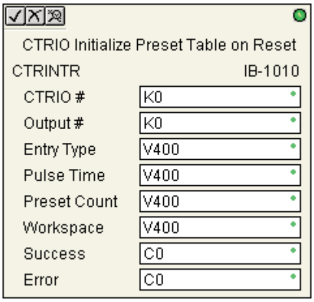
Rung 4: This rung allows the operator to enable output #0 from the ladder code.



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

DS5/6	Used
HPP	N/A

CTRIO Initialize Preset Table on Reset, on a leading edge transition to this IBox, defines the initial preset table to be loaded automatically when the reset event occurs, on a specific CTRIO Output resource.This IBox will take more than 1 PLC scan to execute. Either the Success or Error bit will turn on when the command is complete. If the Error Bit is on, you can use the CTRIO Read Error Code (CTRRDER) IBox to get extended error information.



CTRIO Initialize Preset Table on Reset

CTRINTR IB-1010

CTRIO #	K0
Output #	K0
Entry Type	V400
Pulse Time	V400
Preset Count	V400
Workspace	V400
Success	C0
Error	C0

Entry Type:

K0: Set

K1: Reset

K2: Pulse On (uses Pulse Time)

K3: Pulse Off (uses Pulse Time)

K4: Toggle

K5: Reset Count

Note that the Pulse Time parameter is ignored by some Entry Types.

The Workspace register is for internal use by this IBox instruction and MUST NOT be used anywhere else in your program.

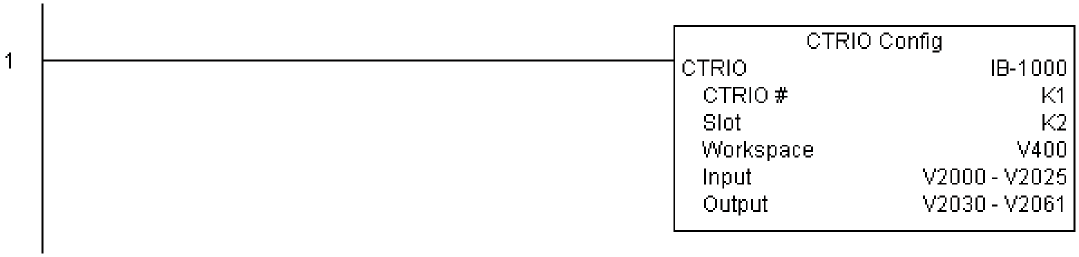
CTRINTR Parameters

- CTRIO#: specifies a specific CTRIO module based on a user defined number (see CTRIO Config Ibox)
- Output#: specifies a CTRIO output to be used by the instruction
- Entry Type: specifies the Entry Type to add during the edit
- Pulse Time: specifies a pulse time for the Pulse On and Pulse Off Entry Types
- Preset Count: specifies an initial count value to begin at after Reset
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the instruction has successfully completed
- Error: specifies a bit that will turn on if the instruction does not complete successfully

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Entry Type . . . . V,K	K0-5; See DL405 V-memory map - Data Words
Pulse Time . . . . V,K	K0-65535; See DL405 V-memory map - Data Words
Preset Count . . . . V,K	K0-2147483647; See DL405 V-memory map
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

CTRINTR Example

Rung 1: This sets up the CTRIO card in slot 2 of the local base. Each CTRIO in the system will need a separate CTRIO I-box before any CTRxxxx I-boxes can be used for them. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2030 through V2061 for its output data.



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

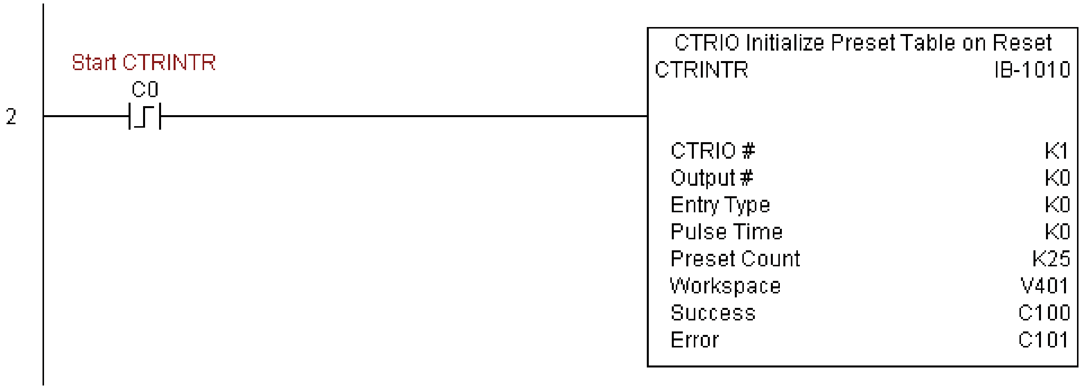
(example continued on next page)

CTRINTR Example (con't)

Rung 2: This rung is a sample method for enabling the CTRINTR command. A C-bit is used to allow the programmer to control the command from Data View for testing purposes.

Turning on C0 will cause the CTRINTR instruction to create a single entry preset table, but not as a file, and use it for output #0, the new preset will be loaded when the current count is reset. In this case the single preset will be a set at a count of 25 for output #0.

Operating procedure for this example code is to load the CTRINTR_ex1.cwb file to your CTRIO, then enter the code shown here, change to RUN mode, enable output #0 by turning on C2 in Data View, turn encoder on CTRIO to value above 10 and output #0 light will come on. Now turn on C0 to execute the CTRINTR command, reset the counter with C1, then turn encoder to value of 25+ (output #0 should turn on).



CTRIO Initialize Preset Table on Reset

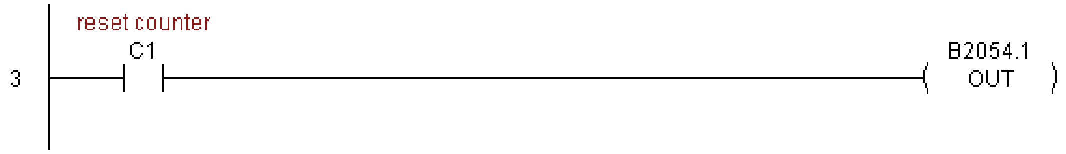
CTRINTR IB-1010

CTRIO #	K1
Output #	K0
Entry Type	K0
Pulse Time	K0
Preset Count	K25
Workspace	V401
Success	C100
Error	C101

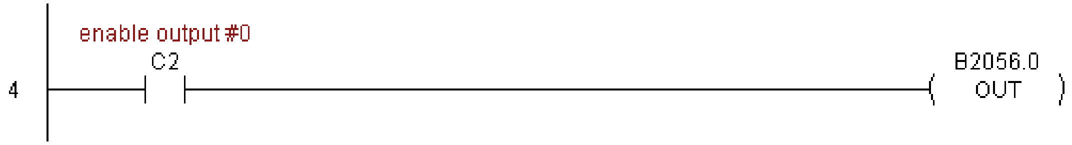
(example continued on next page)

CTRINTR Example (con't)

Rung 3: This rung allows the programmer to reset the counter from the ladder logic.



Rung 4: This rung allows the operator to enable output #0 from the ladder code.

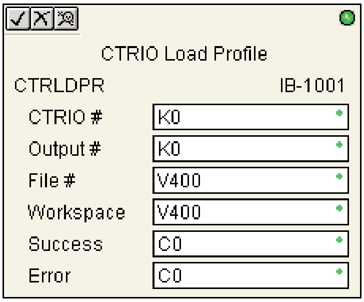


CTRIO Load Profile (CTRLDPR) (IB-1001)

DS5/6	Used
HPP	N/A

CTRIO Load Profile loads a CTRIO Profile File to a CTRIO Output resource on a leading edge transition to this IBox. This IBox will take more than 1 PLC scan to execute. Either the Success or Error bit will turn on when the command is complete. If the Error Bit is on, you can use the CTRIO Read Error Code (CTRRDER) IBox to get extended error information.

The Workspace register is for internal use by this IBox instruction and MUST NOT be used anywhere else in your program.



CTRIO Load Profile

CTRLDPR IB-1001

CTRIO #	K0
Output #	K0
File #	V400
Workspace	V400
Success	C0
Error	C0

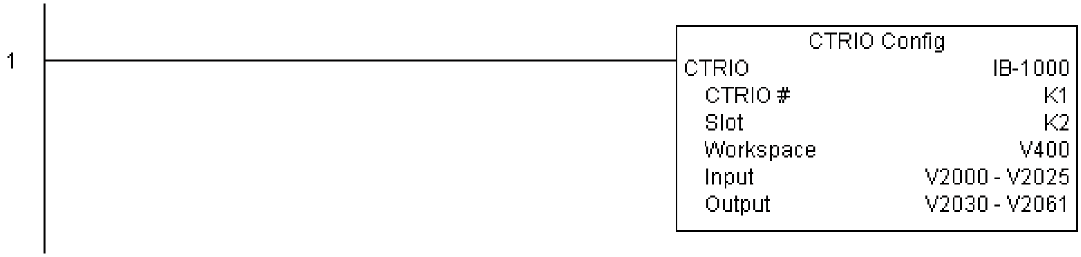
CTRLDPR Parameters

- CTRIO#: specifies a specific CTRIO module based on a user defined number (see CTRIO Config)
- Output#: specifies a CTRIO output to be used by the instruction
- File#: specifies a CTRIO profile File number to be loaded
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the instruction has successfully completed
- Error: specifies a bit that will turn on if the instruction does not complete successfully

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
File# . . . . V,K	K0-255; See DL405 V-memory map - Data Words
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

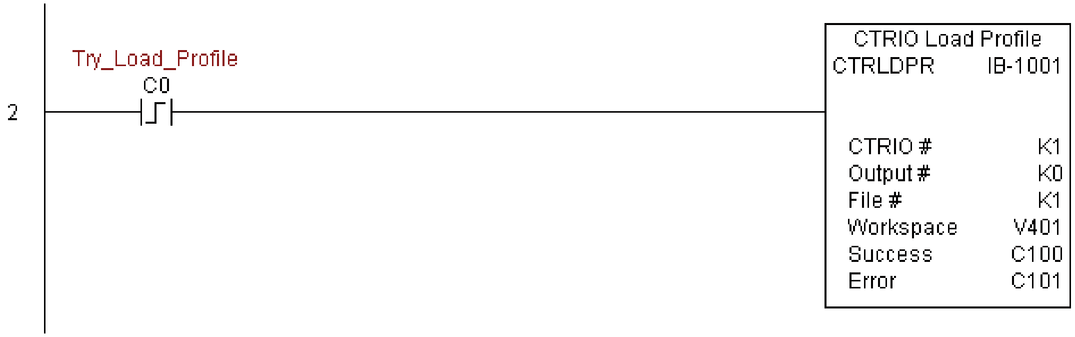
CTRLDPR Example

Rung 1: This sets up the CTRIO card in slot 2 of the local base. Each CTRIO in the system will need a separate CTRIO I-box before any CTRxxxx I-boxes can be used for them. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2030 through V2061 for its output data.



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

Rung 2: This CTRIO Load Profile IBox will load File #1 into the working memory of Output 0 in CTRIO #1. This example program requires that you load CTRLDPR_IBox.cwb into your Hx-CTRIO module.

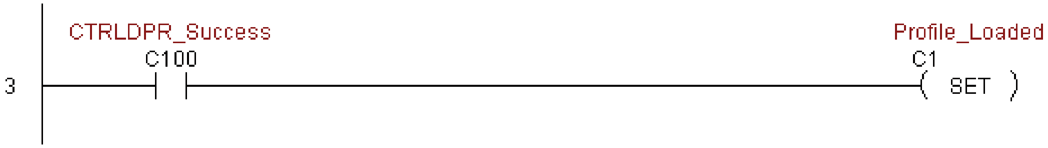


CTRIO Load Profile
CTRLDPR IB-1001
CTRIO # K1
Output # K0
File # K1
Workspace V401
Success C100
Error C101

(example continued on next page)

CTRLDPR Example (con't)

Rung 3: If the file is successfully loaded, set Profile_Loaded.



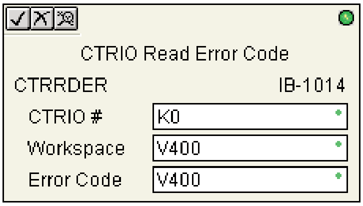
CTRIO Read Error (CTRRDER) (IB-1014)

DS5/6	Used
HPP	N/A

CTRIO Read Error Code will get the decimal error code value from the CTRIO module (listed below) and place it into the given Error Code register, on a leading edge transition to the IBox. This instruction is not supported when the CTRIO is used in an ERM/EBC configuration.

Since the Error Code in the CTRIO is only maintained until another CTRIO command is given, you must use this instruction immediately after the CTRIO IBox that reports an error via its Error bit parameter.

The Workspace register is for internal use by this IBox instruction and MUST NOT be used anywhere else in your program.



CTRIO Read Error Code

CTRRDER IB-1014

CTRIO # K0

Workspace V400

Error Code V400

Error Codes:

0: No Error

100: Specified command code is unknown or unsupported

101: File number not found in the file system

102: File type is incorrect for specified output function

103: Profile type is unknown

104: Specified input is not configured as a limit on this output

105: Specified limit input edge is out of range

106: Specified input function is unconfigured or invalid

107: Specified input function number is out of range

108: Specified preset function is invalid

109: Preset table is full

110: Specified Table entry is out of range

111: Specified register number is out of range

112: Specified register is an unconfigured input or output

2001: Error reading Error Code - cannot access CTRIO via ERM

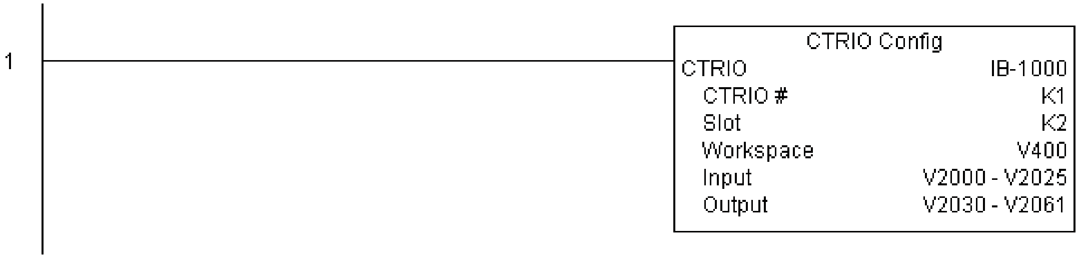
CTRRDER Parameters

- CTRIO#: specifies a specific CTRIO module based on a user defined number (see CTRIO Config)
- Workspace: specifies a V-memory location that will be used by the instruction
- Error Code: specifies the location where the Error Code will be written

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Workspace V	See DL405 V-memory map - Data Words
Error Code V	See DL405 V-memory map - Data Words

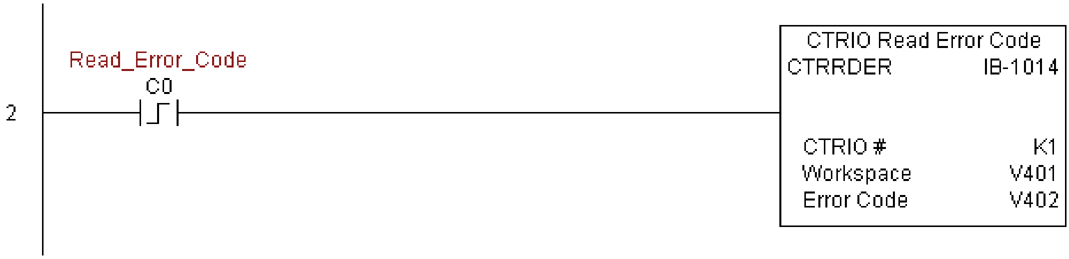
CTRRDER Example

Rung 1: This sets up the CTRIO card in slot 2 of the local base. Each CTRIO in the system will need a separate CTRIO I-box before any CTRxxxx I-boxes can be used for them. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2030 through V2061 for its output data.



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

Rung 2: This CTRIO Read Error Code IBox will read the Extended Error information from CTRIO #1. This example program requires that you load CTRRDER_IBox.cwb into your Hx-CTRIO module.

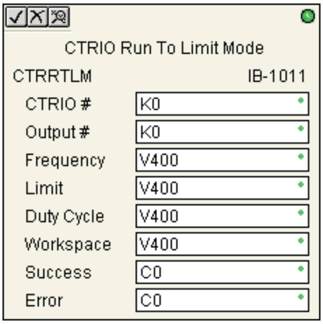


CTRIO Read Error Code
CTRRDER IB-1014
CTRIO # K1
Workspace V401
Error Code V402

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

DS5/6	Used
HPP	N/A

CTRIO Run To Limit Mode, on a leading edge transition to this IBox, loads the Run to Limit command and given parameters on a specific Output resource. The CTRIO's Input(s) must be configured as Limit(s) for this function to work.



CTRIO Run To Limit Mode

CTRRTLM IB-1011

CTRIO #	K0
Output #	K0
Frequency	V400
Limit	V400
Duty Cycle	V400
Workspace	V400
Success	C0
Error	C0

Valid Hexadecimal Limit Values:

K00 - Rising Edge of Ch1/C

K10 - Falling Edge of Ch1/C

K20 - Both Edges of Ch1/C

K01 - Rising Edge of Ch1/D

K11 - Falling Edge of Ch1/D

K21 - Both Edges of Ch1/D

K02 - Rising Edge of Ch2/C

K12 - Falling Edge of Ch2/C

K22 - Both Edges of Ch2/C

K03 - Rising Edge of Ch2/D

K13 - Falling Edge of Ch2/D

K23 - Both Edges of Ch2/D

This IBox will take more than 1 PLC scan to execute. Either the Success or Error bit will turn on when the command is complete. If the Error Bit is on, you can use the CTRIO Read Error Code (CTRRDER) IBox to get extended error information.

The Workspace register is for internal use by this IBox instruction and MUST NOT be used anywhere else in your program.

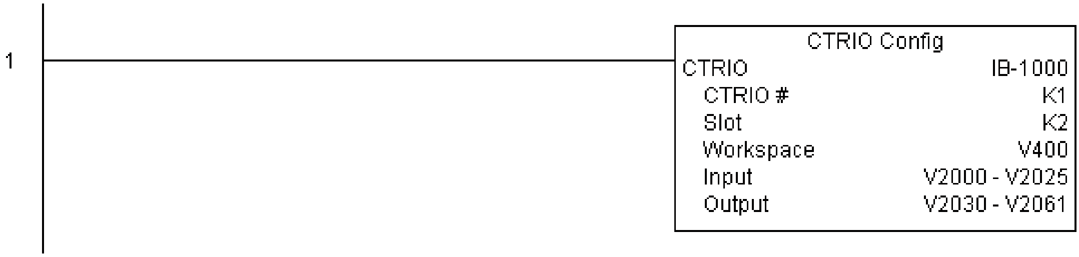
CTRRTLM Parameters

- CTRIO#: specifies a specific CTRIO module based on a user defined number (see CTRIO Config Ibox)
- Output#: specifies a CTRIO output to be used by the instruction
- Frequency: specifies the output pulse rate (20Hz - 20KHz)
- Limit: the CTRIO's Input(s) must be configured as Limit(s) for this function to operate
- Duty Cycle: specifies the % of on time versus off time. This is a hex number. Default of 0 is 50%, also entering 50 will yield 50%. 50% duty cycle is defined as on half the time and off half the time
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the instruction has successfully completed
- Error: specifies a bit that will turn on if the instruction does not complete successfully

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Frequency . . . . V,K	K20-20000; See DL405 V-memory map - Data Words
Limit . . . . V,K	K0-FF; See DL405 V-memory map - Data Words
Duty Cycle . . . . V,K	K0-99; See DL405 V-memory map - Data Words
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

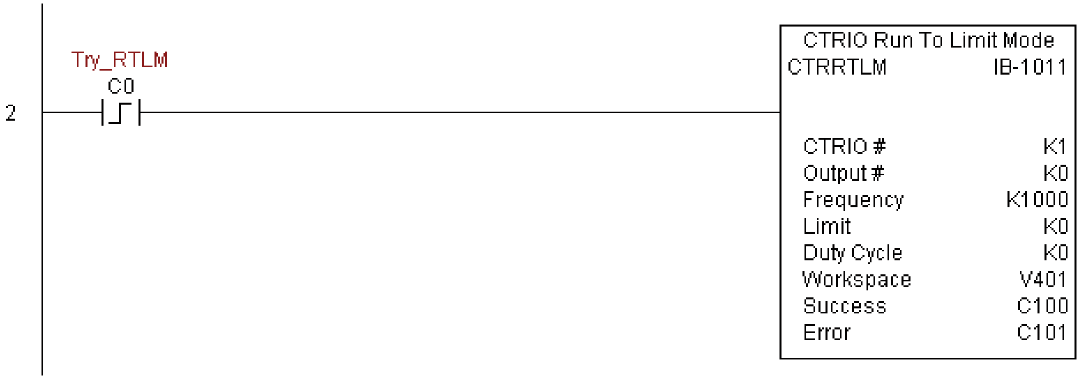
CTRRTLM Example

Rung 1: This sets up the CTRIO card in slot 2 of the local base. Each CTRIO in the system will need a separate CTRIO I-box before any CTRxxxx I-boxes can be used for them. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2030 through V2061 for its output data.



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

Rung 2: This CTRIO Run To Limit Mode IBox sets up Output #0 in CTRIO #1 to output pulses at a Frequency of 1000 Hz until Llimit #0 comes on. This example program requires that you load CTRRTLM_IBox.cwb into your Hx-CTRIO module.

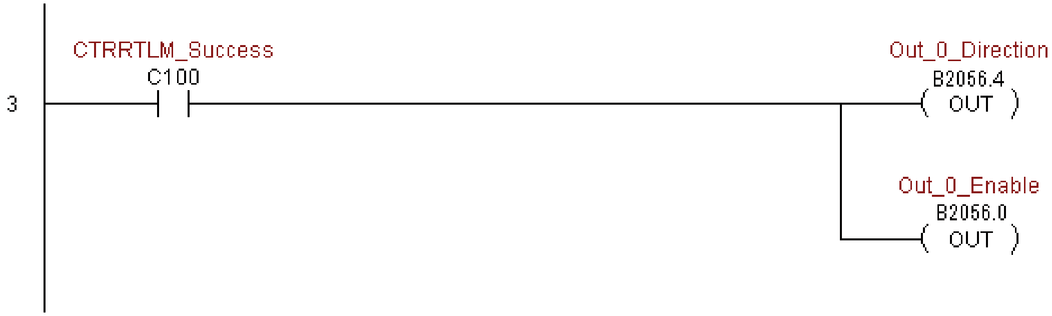


CTRIO Run To Limit Mode
CTRRTLM IB-1011
CTRIO # K1
Output # K0
Frequency K1000
Limit K0
Duty Cycle K0
Workspace V401
Success C100
Error C101

(example continued on next page)

CTRRTLM Example (con't)

Rung 3: If the Run To Limit Mode parameters are OK, set the Direction Bit and Enable the output.



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

DS5/6	Used
HPP	N/A

CTRIO Run To Position Mode, on a leading edge transition to this IBox, loads the Run to Position command and given parameters on a specific Output resource.

Valid Function Values are:

00: Less Than Ch1/Fn1

10: Greater Than Ch1/Fn1

01: Less Than Ch1/Fn2

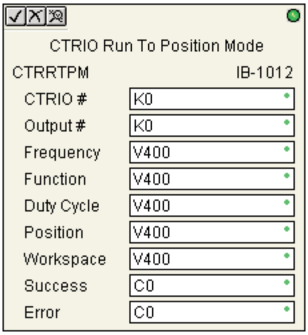
11: Greater Than Ch1/Fn2

02: Less Than Ch2/Fn1

12: Greater Than Ch2/Fn1

03: Less Than Ch2/Fn2

13: Greater Than Ch2/Fn2



CTRIO Run To Position Mode

CTRRTPM IB-1012

CTRIO # K0

Output # K0

Frequency V400

Function V400

Duty Cycle V400

Position V400

Workspace V400

Success C0

Error C0

This IBox will take more than 1 PLC scan to execute. Either the Success or Error bit will turn on when the command is complete. If the Error Bit is on, you can use the CTRIO Read Error Code (CTRRDER) IBox to get extended error information.

The Workspace register is for internal use by this IBox instruction and MUST NOT be used anywhere else in your program.

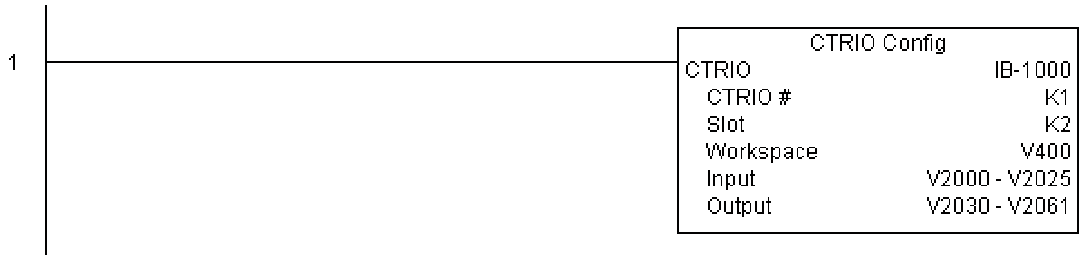
CTRRTPM Parameters

- CTRIO#: specifies a specific CTRIO module based on a user defined number (see CTRIO Config Ibox)
- Output#: specifies a CTRIO output to be used by the instruction
- Frequency: specifies the output pulse rate (20Hz - 20KHz)
- Duty Cycle: specifies the % of on time versus off time. This is a hex number. Default of 0 is 50%, also entering 50 will yield 50%. 50% duty cycle is defined as on half the time and off half the time
- Position: specifies the count value, as measured on the encoder input, at which the output pulse train will be turned off
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the instruction has successfully completed
- Error: specifies a bit that will turn on if the instruction does not complete successfully

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Frequency . . . . V,K	K20-20000; See DL405 V-memory map - Data Words
Duty Cycle . . . . V,K	K0-99; See DL405 V-memory map
Position . . . . V,K	K0-2147483647; See DL405 V-memory map
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

CTRRTPM Example

Rung 1: This sets up the CTRIO card in slot 2 of the local base. Each CTRIO in the system will need a separate CTRIO I-box before any CTRxxxx I-boxes can be used for them. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2030 through V2061 for its output data.

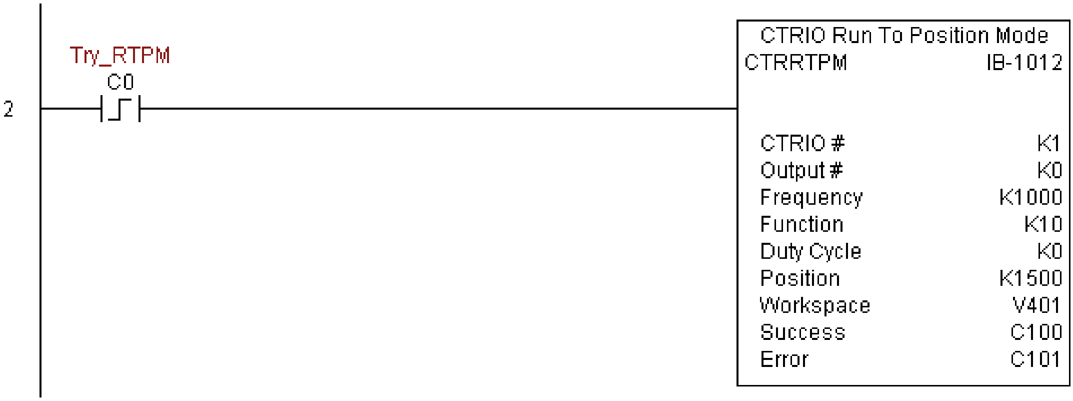


```
CTRIO Config
CTRIO                IB-1000
  CTRIO #                 K1
  Slot                    K2
  Workspace             V400
  Input        V2000 - V2025
  Output       V2030 - V2061
```

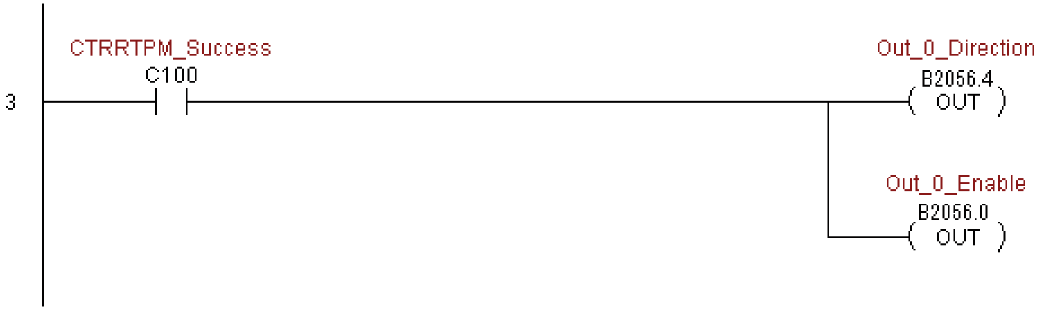
(example continued on next page)

CTRRTPM Example (con't)

Rung 2: This CTRIO Run To Position Mode IBox sets up Output #0 in CTRIO #1 to output pulses at a Frequency of 1000 Hz, use the 'Greater than Ch1/Fn1' comparison operator, until the input position of 1500 is reached. This example program requires that you load CTRRTPM_IBox.cwb into your Hx-CTRIO module.



Rung 3: If the Run To Position Mode parameters are OK, set the Direction Bit and Enable the output.



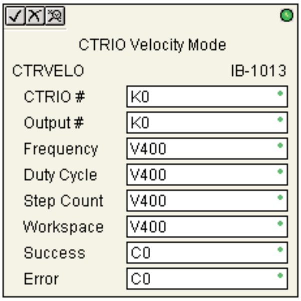
CTRIO Velocity Mode (CTRVELO) (IB-1013)

DS5/6	Used
HPP	N/A

CTRIO Velocity Mode loads the Velocity command and given parameters on a specific Output resource on a leading edge transition to this IBox.

This IBox will take more than 1 PLC scan to execute. Either the Success or Error bit will turn on when the command is complete. If the Error Bit is on, you can use the CTRIO Read Error Code (CTRRDER) IBox to get extended error information.

The Workspace register is for internal use by this IBox instruction and MUST NOT be used anywhere else in your program.



CTRIO Velocity Mode

CTRVELO IB-1013

CTRIO #	K0
Output #	K0
Frequency	V400
Duty Cycle	V400
Step Count	V400
Workspace	V400
Success	C0
Error	C0

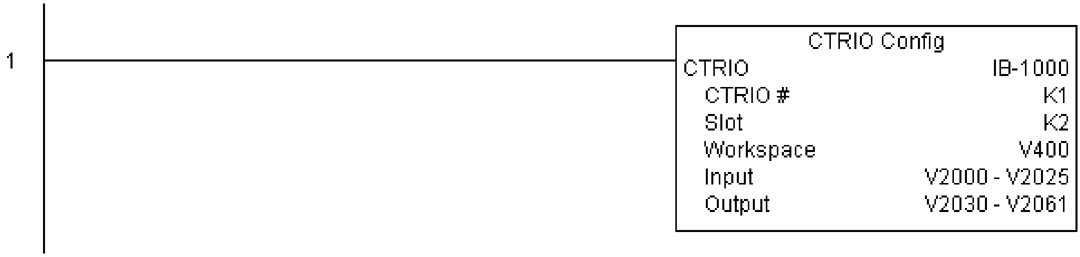
CTRVELO Parameters

- CTRIO#: specifies a specific CTRIO module based on a user defined number (see CTRIO Config Ibox)
- Output#: specifies a CTRIO output to be used by the instruction
- Frequency: specifies the output pulse rate (20Hz - 20KHz)
- Duty Cycle: specifies the % of on time versus off time. This is a hex number. Default of 0 is 50%, also entering 50 will yield 50%. 50% duty cycle is defined as on half the time and off half the time
- Step Count: specifies the number of pulses to output as a 32-bit Hex number, a value of Kffffffff will cause the profile to run continuously as long as the output is enabled
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the instruction has successfully completed
- Error: specifies a bit that will turn on if the instruction does not complete successfully

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Frequency . . . . V,K	K20-20000; See DL405 V-memory map - Data Words
Duty Cycle . . . . V,K	K0-99; See DL405 V-memory map
Step Count . . . . V,K	K0-2147483647; See DL405 V-memory map
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

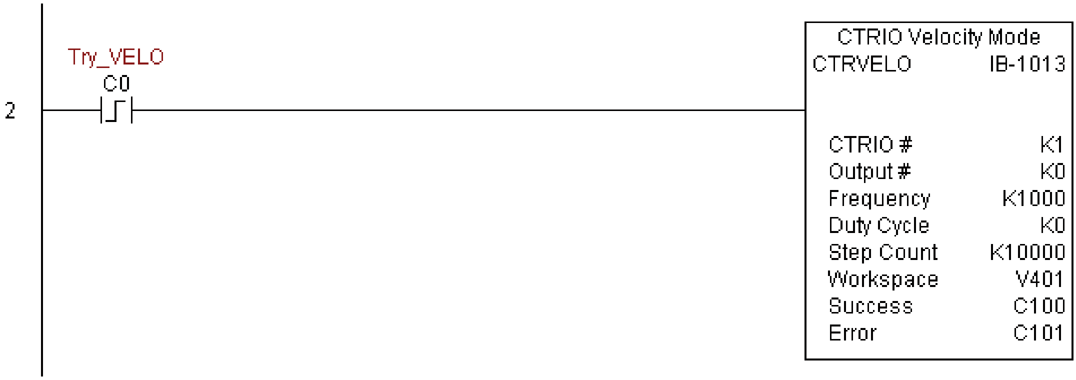
CTRVELO Example

Rung 1: This sets up the CTRIO card in slot 2 of the local base. Each CTRIO in the system will need a separate CTRIO I-box before any CTRxxxx I-boxes can be used for them. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2030 through V2061 for its output data.



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

Rung 2: This CTRIO Velocity Mode IBox sets up Output #0 in CTRIO #1 to output 10,000 pulses at a Frequency of 1000 Hz. This example program requires that you load CTRVELO_IBox.cwb into your Hx-CTRIO module.

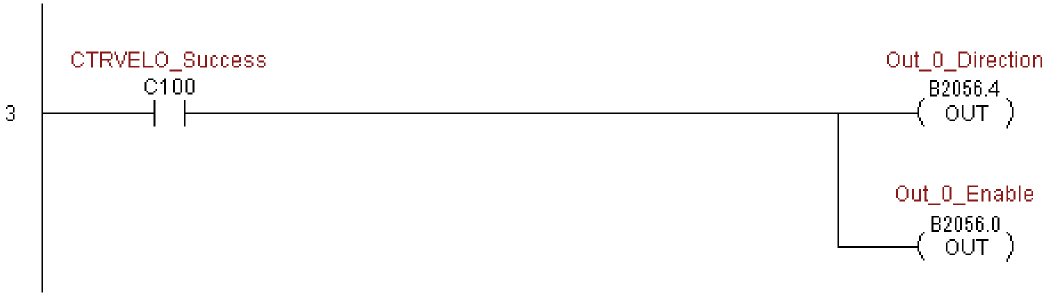


CTRIO Velocity Mode
CTRVELO IB-1013
CTRIO # K1
Output # K0
Frequency K1000
Duty Cycle K0
Step Count K10000
Workspace V401
Success C100
Error C101

(example continued on next page)

CTRVELO Example (con't)

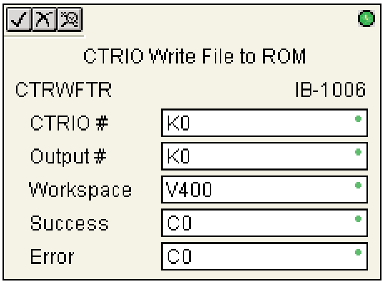
Rung 3: If the Velocity Mode parameters are OK, set the Direction Bit and Enable the output.



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

DS5/6	Used
HPP	N/A

CTRIO Write File to ROM writes the runtime changes made to a loaded CTRIO Preset Table back to Flash ROM on a leading edge transition to this IBox. Writing Preset Table changes to ROM can prevent them from being lost during a power cycle. This IBox will take more than 1 PLC scan to execute. Either the Success or Error bit will turn on when the command is complete. If the Error Bit is on, you can use the CTRIO Read Error Code (CTRRDER) IBox to get extended error information.



CTRIO Write File to ROM

CTRWFTR IB-1006

CTRIO # K0

Output # K0

Workspace V400

Success C0

Error C0

The Workspace register is for internal use by this IBox instruction and MUST NOT be used anywhere else in your program.

CTRWFTR Parameters

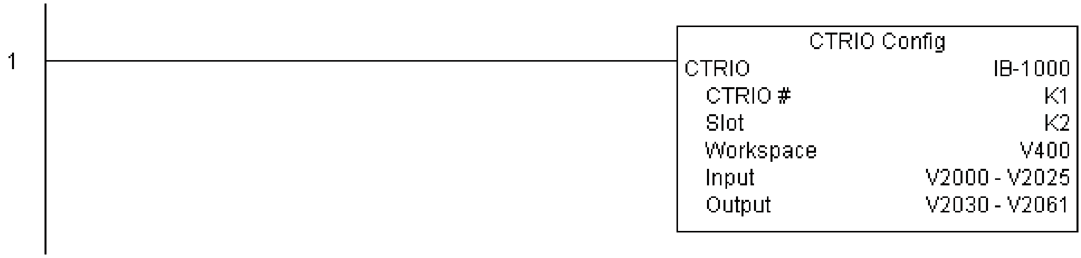
- CTRIO#: specifies a specific CTRIO module based on a user defined number (see CTRIO Config Ibox)
- Output#: specifies a CTRIO output to be used by the instruction
- Workspace: specifies a V-memory location that will be used by the instruction
- Success: specifies a bit that will turn on once the instruction has successfully completed
- Error: specifies a bit that will turn on if the instruction does not complete successfully

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Workspace V	See DL405 V-memory map - Data Words
Success . . . . X,Y,C,GX,GY,B	See DL405 V-memory map
Error . . . . X,Y,C,GX,GY,B	See DL405 V-memory map

Note: *Writing preset tables to ROM on a continual basis can cause the FLASH memory to fail over time. It is recommended only to write these tables to ROM when needed.*

CTRWFTR Example

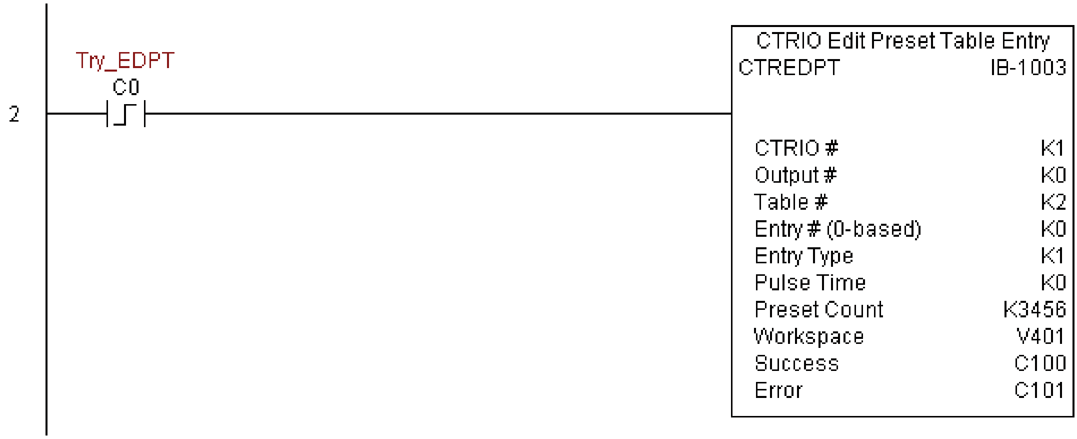
Rung 1: This sets up the CTRIO card in slot 2 of the local base. Each CTRIO in the system will need a separate CTRIO I-box before any CTRxxxx I-boxes can be used for them. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2030 through V2061 for its output data.



CTRIO Config

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

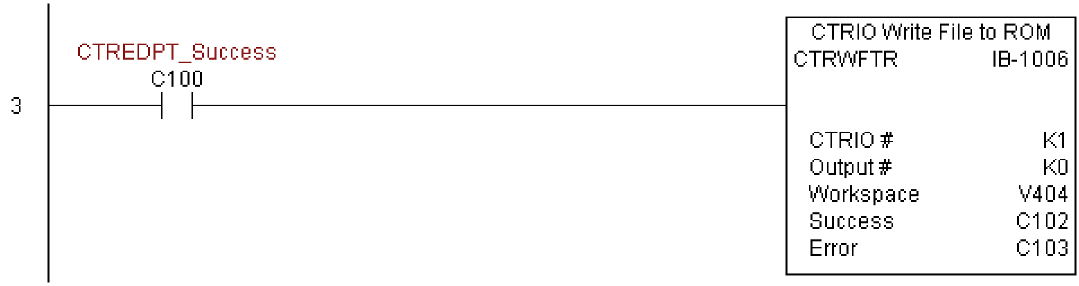
Rung 2: This CTRIO Edit Preset Table Entry IBox will change Entry 0 in Table #2 to be a RESET at Count 3456. This example program requires that you load CTRWFTR_IBox.cwb into your Hx-CTRIO module.



CTRIO Edit Preset Table Entry

CTREDPT	IB-1003
CTRIO #	K1
Output #	K0
Table #	K2
Entry # (0-based)	K0
Entry Type	K1
Pulse Time	K0
Preset Count	K3456
Workspace	V401
Success	C100
Error	C101

Rung 3: If the file is successfully editted, use a Write File To ROM IBox to save the edited table back to the CTRIO's ROM, thereby making the changes retentive.



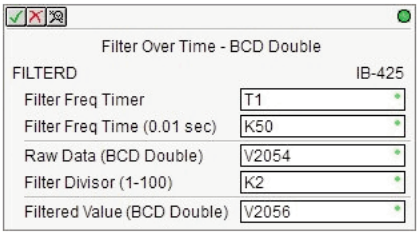
CTRIO Write File to ROM

CTRWFTR	IB-1006
CTRIO #	K1
Output #	K0
Workspace	V404
Success	C102
Error	C103

Filter Over Time - BCD Double (FILTERD) (IB-425)

DS6 **ONLY**	Used
HPP	N/A

The Filter Over Time - BCD Double IBox performs a first-order filter on the specified 32-bit Raw BCD Data value using the specified time interval.



Filter Over Time - BCD Double

FILTERD	IB-425
Filter Freq Timer	T1
Filter Freq Time (0.01 sec)	K50
Raw Data (BCD Double)	V2054
Filter Divisor (1-100)	K2
Filtered Value (BCD Double)	V2056

A first order is essentially a lag function, so the FDC (Filter Divisor Constant) represents the amount of desired lag. A Value of 1 represents no lag, a value of 100 represents the maximum amount of lag.

The formula used is:

$$New = Old + \frac{\left[(Raw - Old) + \left(\frac{FDC}{2}\right)\right]}{FDC}$$

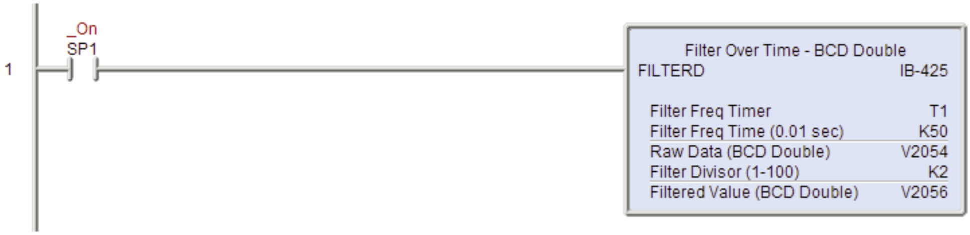
FILTERD Parameters

- Filter Freq Timer: The PLC Timer used to generate the calculation time intervals.
- Filter Freq Time (0.01 sec): The timer preset value in tens of milliseconds (BCD) which specifies the rate at which the calculations take place.
- Raw Data (BCD Double): The first V-Memory of two successive V-Memory locations where the 32-bit BCD input data value is stored.
- Filter Divisor: This value specifies the amount of desired lag (BCD).
- Filter Value (BCD Double): The first V-Memory of two successive V-Memory locations where the new 32-bit filtered output value will be stored.

Parameter	DL405 Range
Filter Freq Timer . . . . T	T0-T377
Filter Freq Time . . . . V, K	K0-9999, All V Memory
Raw Data V	All V Memory
Filter Divisor . . . . V, K	K1-100, All V Memory
Filter Value V	All V Memory

FILTERD Example

In the following example, the FILTERD instruction is used to filter a double word BCD value that is in V2054-V2055. Timer(T1) is set to 0.5 sec, the rate at which the filter calculation will be performed. The filter constant is set to 2. A larger value will increase the smoothing effect of the filter. A value of 1 results in no filtering. The filtered value will be placed in V2056-V2057.



Filter Over Time - BCD Double
FILTERD IB-425

Filter Freq Timer T1
Filter Freq Time (0.01 sec) K50
Raw Data (BCD Double) V2054
Filter Divisor (1-100) K2
Filtered Value (BCD Double) V2056

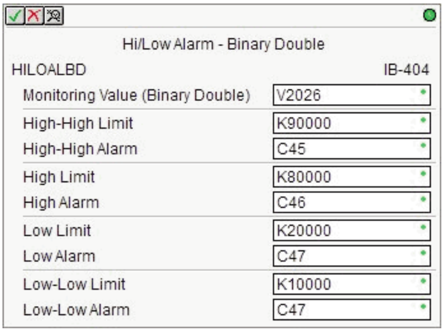
Hi/Lo Alarm - Binary Double (HILOALBD) (IB-404)

DS6 **ONLY**	Used
HPP	N/A

The Hi/Lo Alarm - Binary Double IBox monitors the 32-bit binary (decimal) value that is stored in two successive V-Memory locations and sets the appropriate alarm states based on the alarm limit values.

When you enter the alarm limit values you must ensure that the High-High limit ≥ the High limit ≥ the Low limit ≥ the Low-Low limit.

The alarm limits are inclusive. For example, the High and High-High alarm bits will be ON when the Monitoring Value ≥ High-High limit and the Monitoring Value ≥ High limit. The Low and Low-Low alarm bits will be ON when the Monitoring Value ≤ Low limit and the Monitoring Value ≤ Low-Low limit.



Hi/Low Alarm - Binary Double

HILOALBD	IB-404
Monitoring Value (Binary Double)	V2026
High-High Limit	K90000
High-High Alarm	C45
High Limit	K80000
High Alarm	C46
Low Limit	K20000
Low Alarm	C47
Low-Low Limit	K10000
Low-Low Alarm	C47

HILOALBD Parameters

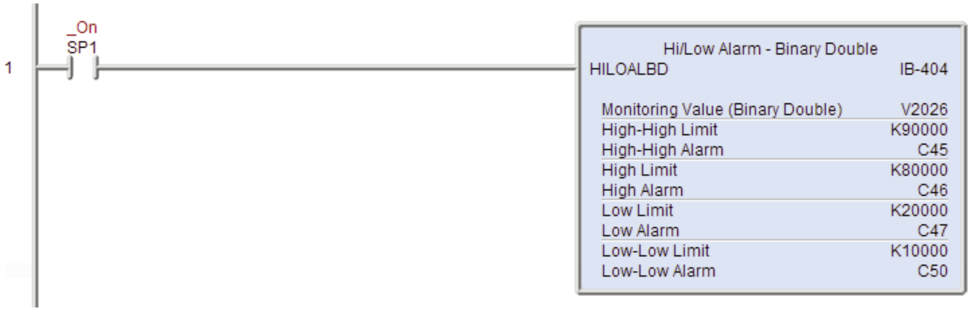
- Monitoring Value (Binary Double): The first V-Memory location of the 32-bit binary (decimal) value to monitor.
- High-High Limit: The High-High alarm limit value (binary double).
- High-High Alarm: The High-High alarm output BIT.
- High Limit: The High alarm limit value (binary double).
- High Alarm: The High alarm output BIT.
- Low Limit: The Low alarm limit value (binary double).
- Low Alarm: The Low alarm output BIT.
- Low-Low Limit: The Low-Low alarm limit value (binary double).
- Low-Low Alarm: The Low-Low alarm output BIT.

Parameter	DL405 Range
Monitoring Value V	All V Memory
High-High Limit . . . . V, K	K0-4294967295; All V Memory
High-High Alarm . . . . X, Y, C, GX,GY, B	All Bit Memory
High Limit . . . . V, K	K0-4294967295; All V Memory
High Alarm . . . . X, Y, C, GX,GY, B	All Bit Memory
Low Limit . . . . V, K	K0-4294967295; All V Memory
Low Alarm . . . . X, Y, C, GX,GY,B	All Bit Memory
Low-Low Limit . . . . V, K	K0-4294967295; All V Memory
Low-Low Alarm. . . . . X, Y, C, GX,GY, B	All Bit Memory

HILOALBD Example

In the following example, the HILOALBD instruction is used to monitor a double word binary value that is in V2026-V2027. If the value in V2026-V2027 meets/exceeds the high limit of K80000, C46 will turn ON. If the value continues to increase to meet/exceed the high-high limit of K90000, C45 will turn ON. Both bits would be ON in this case. The high and high-high limits and alarms can be set to the same value if one “high” limit or alarm is desired to be used.

If the value in V2026-V2027 meets or falls below the low limit of K20000, C47 will turn ON. If the value continues to decrease to meet or fall below the low-low limit of K10000, C50 will turn ON. Both bits would be ON in this case. The low and low-low limits and alarms can be set to the same value if one “low” limit or alarm is desired to be used.



Hi/Low Alarm - Binary Double

HILOALBD IB-404

Monitoring Value (Binary Double)	V2026
High-High Limit	K90000
High-High Alarm	C45
High Limit	K80000
High Alarm	C46
Low Limit	K20000
Low Alarm	C47
Low-Low Limit	K10000
Low-Low Alarm	C50

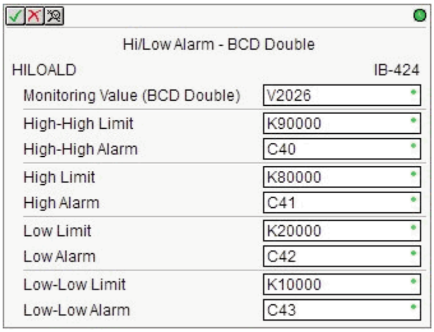
Hi/Lo Alarm - BCD Double (HILOALD) (IB-424)

DS6 ONLY	Used
HPP	N/A

The Hi/Lo Alarm - BCD Double IBox monitors the 32-bit BCD value that is stored in two successive V-Memory locations and sets the appropriate alarm states based on the alarm limit values.

When you enter the alarm limit values you must ensure that the High-High limit ≥ the High limit ≥ the Low limit ≥ the Low-Low limit.

The alarm limits are inclusive. For example, the High and High-High alarm bits will be ON when the Monitoring Value ≥ High-High limit and the Monitoring Value ≥ High limit. The Low and Low-Low alarm bits will be ON when the Monitoring Value ≤ Low limit and the Monitoring Value ≤ Low-Low limit.



Hi/Low Alarm - BCD Double

HILOALD IB-424

Monitoring Value (BCD Double)	V2026
High-High Limit	K90000
High-High Alarm	C40
High Limit	K80000
High Alarm	C41
Low Limit	K20000
Low Alarm	C42
Low-Low Limit	K10000
Low-Low Alarm	C43

HILOALD Parameters

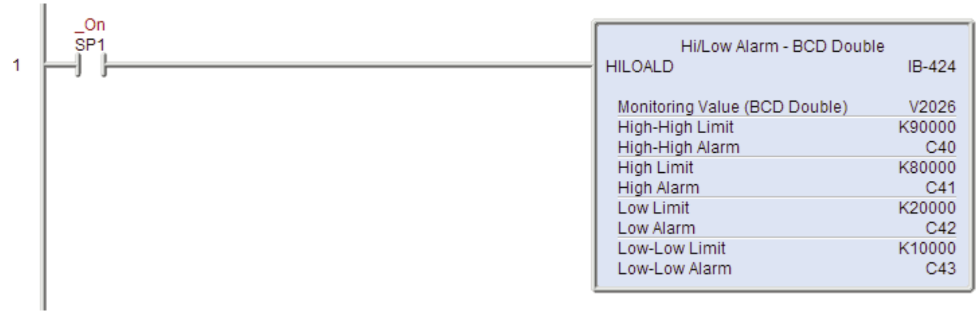
- Monitoring Value (BCD Double): The first V-Memory location of the 32-bit BCD value to monitor.
- High-High Limit: The High-High alarm limit value (BCD double).
- High-High Alarm: The High-High alarm output BIT.
- High Limit: The High alarm limit value (BCD double).
- High Alarm: The High alarm output BIT.
- Low Limit: The Low alarm limit value (BCD double).
- Low Alarm: The Low alarm output BIT.
- Low-Low Limit: The Low-Low alarm limit value (BCD double).
- Low-Low Alarm: The Low-Low alarm output BIT.

Parameter	DL405 Range
Monitoring Value V	All V Memory
High-High Limit . . . . V, K	K0-99999999; All V Memory
High-High Alarm . . . . X, Y, C, GX,GY, B	All Bit Memory
High Limit . . . . V, K	K0-99999999; All V Memory
High Alarm . . . . X, Y, C, GX,GY, B	All Bit Memory
Low Limit . . . . V, K	K0-99999999; All V Memory
Low Alarm . . . . X, Y, C, GX,GY,B	All Bit Memory
Low-Low Limit . . . . V, K	K0-99999999; All V Memory
Low-Low Alarm. . . . . X, Y, C, GX,GY, B	All Bit Memory

HILOALD Example

In the following example, the HILOALD instruction is used to monitor a double word BCD value that is in V2026-V2027. If the value in V2026-V2027 meets/exceeds the high limit of K80000, C41 will turn ON. If the value continues to increase to meet/exceed the high-high limit of K90000, C40 will turn ON. Both bits would be ON in this case. The high and high-high limits and alarms can be set to the same value if one “high” limit or alarm is desired to be used.

If the value in V2026-V2027 meets or falls below the low limit of K20000, C42 will turn ON. If the value continues to decrease to meet or fall below the low-low limit of K10000, C43 will turn ON. Both bits would be ON in this case. The low and low-low limits and alarms can be set to the same value if one “low” limit or alarm is desired to be used.



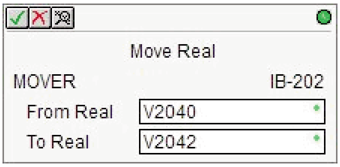
Hi/Low Alarm - BCD Double

HILOALD	IB-424
Monitoring Value (BCD Double)	V2026
High-High Limit	K90000
High-High Alarm	C40
High Limit	K80000
High Alarm	C41
Low Limit	K20000
Low Alarm	C42
Low-Low Limit	K10000
Low-Low Alarm	C43

Move Real (MOVER) (IB-202)

DS6 **ONLY**	Used
HPP	N/A

The Move Real IBox will copy a 32-bit floating point number that is stored in two consecutive V-Memory locations to the specified location which is also two consecutive V-Memory locations.



Move Real

MOVER IB-202

From Real: V2040

To Real: V2042

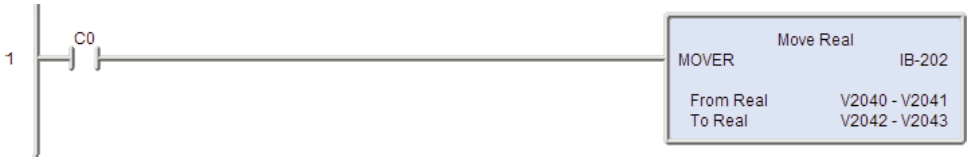
MOVER Parameters

- From Real: The first V-Memory location of the source data double-word.
- To Real: The first V-Memory location of the destination double-word.

Parameter	DL405 Range
From Real . . . . V,R	R-3.402823E+38 - +3.402823E+38; All V Memory
To Real V	All V Memory

MOVER Example

In the following example, the MOVER instruction is used to move 32 bits of data from V2040-V2041 to V2042-V2043.



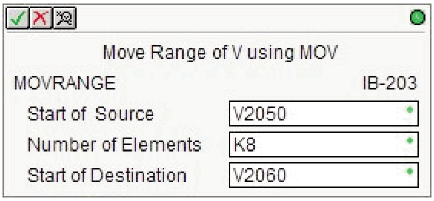
Move Real
MOVER IB-202

From Real V2040 - V2041
To Real V2042 - V2043

Move Range of V using MOV (MOVRANGE) (IB-203)

DS6 **ONLY**	Used
HPP	N/A

The Move Range of V using MOV will use a MOV instruction to copy the values from one range of V-Memory locations to a second range of V-Memory locations. Up to 4095 V-Memory locations can be moved.


Move Range of V using MOV

MOVRANGE IB-203

Start of Source: V2050

Number of Elements: K8

Start of Destination: V2060

MOVRANGE Parameters

- Start of Source: The first V-Memory location of the source range.
- Number of Elements: The number of consecutive V-Memory locations to process (BCD).
- Start of Destination: The first V-Memory location of the destination range.

Parameter	DL405 Range
Start of Source V	All V Memory
Number of Elements . . . . V,K	K1 - 4095, All V Memory
Start of Destination V	All V Memory

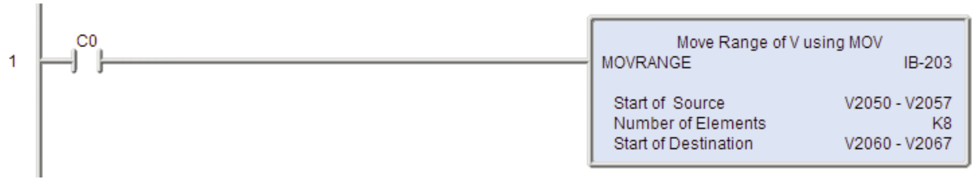
Note: *The Source Range and the Destination Range CAN NOT overlap.*

Note: *If the instruction will be moving double-word values the Number of Elements must be an even number.*

Note: *All of the locations will be moved in the same PLC scan, which will cause an increase in the scan time. Be aware this increase may be large enough to trip with watchdog timer.*

MOVRANGE Example

In the following example, the MOVRANGE instruction is used to move 8 words of data from V2050-V2057 to V2060-V2067.



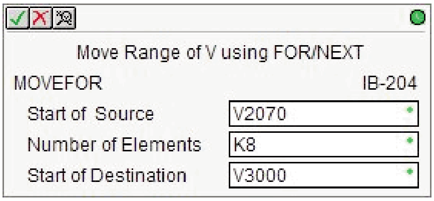
Move Range of V using MOV
MOVRANGE IB-203

Start of Source	V2050 - V2057
Number of Elements	K8
Start of Destination	V2060 - V2067

Move Range of V using FOR/NEXT (MOVEFOR) (IB-204)

DS6 ONLY	Used
HPP	N/A

The Move Range of V using FOR/NEXT will use a FOR/NEXT loop to copy the values from one range of V-Memory locations to a second range of V-Memory locations. Up to 4095 V-Memory locations can be moved.


Move Range of V using FOR/NEXT

MOVEFOR IB-204

Start of Source V2070

Number of Elements K8

Start of Destination V3000

MOVEFOR Parameters

- Start of Source: The first V-Memory location of the source range.
- Number of Elements: The number of consecutive V-Memory locations to process (BCD).
- Start of Destination: The first V-Memory location of the destination range.

Parameter	DL405 Range
Start of Source V	All V Memory
Number of Elements . . . . V,K	K1 - 4095, All V Memory
Start of Destination V	All V Memory

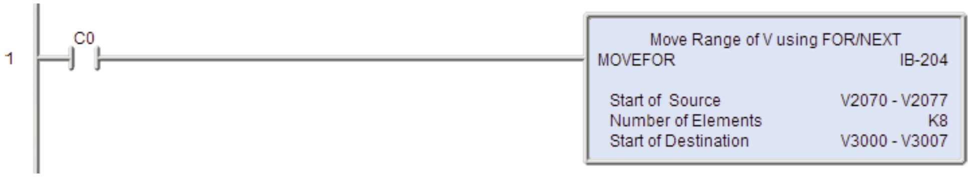
Note: *The Source Range and the Destination Range CAN NOT overlap.*

Note: *If the instruction will be moving double-word values the Number of Elements must be an even number.*

Note: *All of the locations will be moved in the same PLC scan, which will cause an increase in the scan time. Be aware this increase may be large enough to trip with watchdog timer.*

MOVEFOR Example

In the following example, the MOVEFOR instruction is used to move 8 words of data from V2070-V2077 to V3000-V3007.



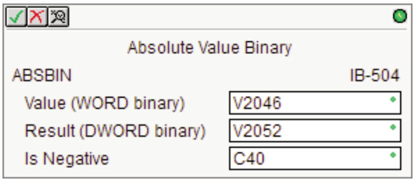
Move Range of V using FOR/NEXT
MOVEFOR IB-204

Start of Source V2070 - V2077
Number of Elements K8
Start of Destination V3000 - V3007

Absolute Value - Binary (ABSBIN) (IB-504)

DS6 **ONLY**	Used
HPP	N/A

The Absolute Value - Binary IBox returns the absolute value of the number Binary (decimal) found in the specified V-Memory location. If the Value is negative, it negates the Value to make it positive and stores it in Result and turns the Is Negative bit ON. Otherwise, it returns the Value unchanged and the Is Negative bit is OFF.



Absolute Value Binary

ABSBIN IB-504

Value (WORD binary)	V2046
Result (DWORD binary)	V2052
Is Negative	C40

For example:

If V2046 = 31415 the result in V2052/V2053 would be 31415, and the Is Negative bit (C40) would be OFF.
If V2046 = -31415 the result in V2052/V2053 would be 31415, and the Is Negative bit (C40) would be ON.

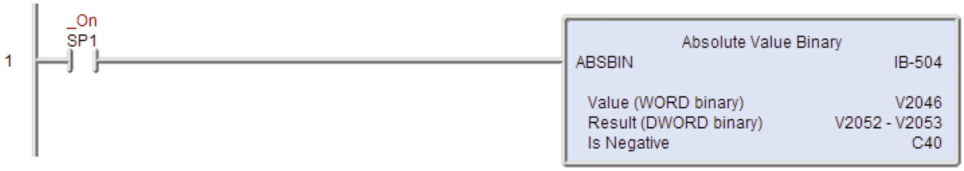
ABSBIN Parameters

- Value (WORD Binary): The V-Memory location where the 16-bit Binary (decimal) value is located.
- Result (DWORD Binary): The first V-Memory location where the 32-bit Binary (decimal) absolute value will be stored.
- Is Negative: If Value (WORD binary) is negative this bit will be ON. If Value (WORD binary) is not negative (e.g. zero or positive) this bit will be OFF.

Parameter	DL405 Range
Value V	All V Memory
Result V	All V Memory
Is Negative . . . . X,Y,C,GX,GY,B	All Bit Memory

ABSBIN Example

In this example the ABSBIN instruction is used to calculate the absolute value of the binary number stored in V2046. The result is stored in V2052-V2053 and C40 will be set if the value of V2046 was negative.



Absolute Value Binary
ABSBIN IB-504

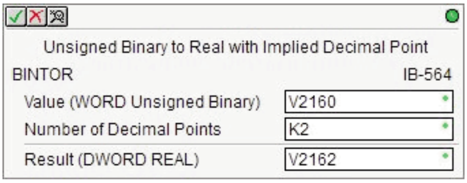
Value (WORD binary) V2046
Result (DWORD binary) V2052 - V2053
Is Negative C40

Unsigned Binary to Real with Implied Decimal Point (BINTOR) (IB-564)

DS6 **ONLY**	Used
HPP	N/A

The Unsigned Binary to Real with Implied Decimal Point IBox converts the given 16-bit Unsigned Binary (decimal) value to a 32-bit real number, given an implied number of decimal points.

Example: K1234 with the Number of Decimal Points set to K1 would yield R123.4.



Unsigned Binary to Real with Implied Decimal Point

BINTOR IB-564

Value (WORD Unsigned Binary) V2160

Number of Decimal Points K2

Result (DWORD REAL) V2162

BINTOR Parameters

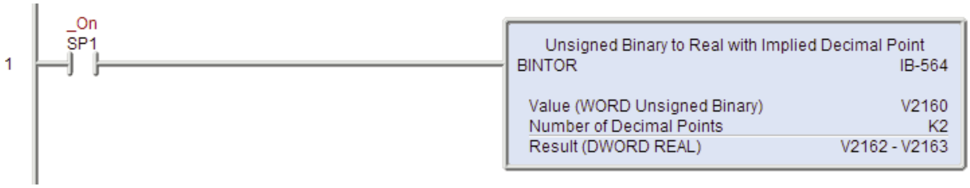
- Value (WORD Unsigned Binary): The V-Memory location where the Unsigned Binary (decimal) value is located, or the constant value to convert.
- Number of Decimal Points: The desired number of digits to the right of the decimal point in the result.
- Result (DWORD Real): The first V-Memory location where the 32-bit Real (floating point) result will be stored.

Parameter	DL405 Range
Value . . . . V,K,P	K0 - 65535, All P Memory, All User V Memory
Number of Decimal Points. . . . . K	K0 - 5
Result V	All User V Memory

BINTOR Example

In the following example the BINTOR instruction is used to convert the binary value stored in V2160 to a 32 bit real number which is stored in V2162-V2163.

K2 in the decimal points implies that the data will have two digits to the right of the decimal point.



Unsigned Binary to Real with Implied Decimal Point
BINTOR IB-564

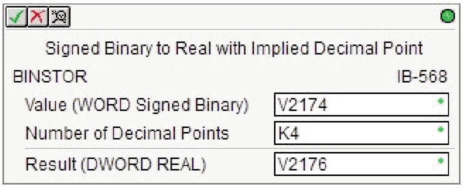
Value (WORD Unsigned Binary) V2160
Number of Decimal Points K2
Result (DWORD REAL) V2162 - V2163

Signed Binary to Real with Implied Decimal Point (BINSTOR) (IB-568)

DS6 **ONLY**	Used
HPP	N/A

The Signed Binary to Real with Implied Decimal Point IBox converts the given 16-bit Signed Binary (decimal) value to a 32-bit real number, given an implied number of decimal points.

Example: K1234 with the Number of Decimal Points set to K1 would yield R123.4.


Signed Binary to Real with Implied Decimal Point
BINSTOR IB-568
Value (WORD Signed Binary) V2174
Number of Decimal Points K4
Result (DWORD REAL) V2176

BINSTOR Parameters

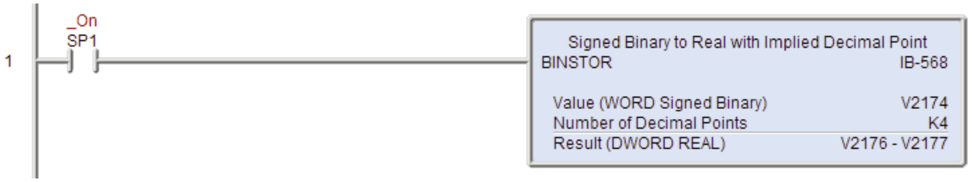
- Value (WORD Signed Binary): The V-Memory location where the Signed Binary (decimal) value is located, or the constant value to convert.
- Number of Decimal Points: The desired number of digits to the right of the decimal point in the result.
- Result (DWORD Real): The first V-Memory location where the 32-bit Real (floating point) result will be stored.

Parameter	DL405 Range
Value . . . . V,K,P	K0 - 32767, All P Memory, All User V Memory
Number of Decimal Points. . . . . K	K0 - 5
Result V	All User V Memory

BINSTOR Example

In the following example the BINSTOR instruction is used to convert the signed binary value stored in V2174 to a 32 bit real number which is stored in V2176-V2177.

K4 in the decimal points implies that the data will have four digits to the right of the decimal point.



Signed Binary to Real with Implied Decimal Point
BINSTOR IB-568

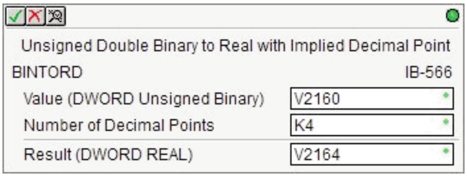
Value (WORD Signed Binary) V2174
Number of Decimal Points K4
Result (DWORD REAL) V2176 - V2177

Unsigned Double Binary to Real with Implied Decimal Point (BINTORD) (IB-566)

DS6 **ONLY**	Used
HPP	N/A

The Unsigned Double Binary to Real with Implied Decimal Point IBox converts the given 32-bit Unsigned Binary (decimal) value to a 32-bit real number, given an implied number of decimal points.

Example: K12345678 with the Number of Decimal Points set to K5 would yield R123.45678.



Unsigned Double Binary to Real with Implied Decimal Point
BINTORD IB-566
Value (DWORD Unsigned Binary) V2160
Number of Decimal Points K4
Result (DWORD REAL) V2164

BINTORD Parameters

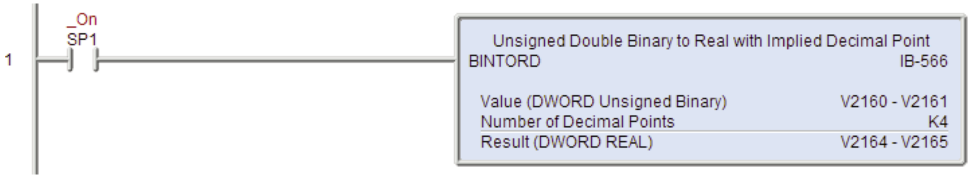
- Value (DWORD Unsigned Binary): The first V-Memory location where the 32-bit Unsigned Binary (decimal) value is located, or the constant value to convert.
- Number of Decimal Points: The desired number of digits to the right of the decimal point in the result.
- Result (DWORD Real): The first V-Memory location where the 32-bit Real (floating point) result will be stored.

Parameter	DL405 Range
Value . . . . V,K,P	K0 - 4294967295, All P Memory, All User V Memory
Number of Decimal Points. . . . . K	K0 - 10
Result V	All User V Memory

BINTORD Example

In the following example the BINTORD instruction is used to convert the double word binary value stored in V2160-V2161 to a 32 bit real number which is stored in V2164-V2165.

K4 in the decimal points implies that the data will have four digits to the right of the decimal point.



Unsigned Double Binary to Real with Implied Decimal Point
BINTORD IB-566

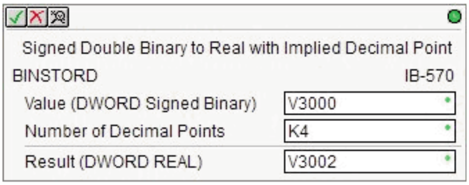
Value (DWORD Unsigned Binary) V2160 - V2161
Number of Decimal Points K4
Result (DWORD REAL) V2164 - V2165

Signed Double Binary to Real with Implied Decimal Point (BINSTORD) (IB-570)

DS6 **ONLY**	Used
HPP	N/A

The Signed Double Binary to Real with Implied Decimal Point IBox converts the given 32-bit Signed Binary (decimal) value to a 32-bit real number, given an implied number of decimal points.

Example: K12345678 with the Number of Decimal Points set to K5 would yield R123.45678.


Signed Double Binary to Real with Implied Decimal Point
BINSTORD IB-570
Value (DWORD Signed Binary) V3000
Number of Decimal Points K4
Result (DWORD REAL) V3002

BINSTORD Parameters

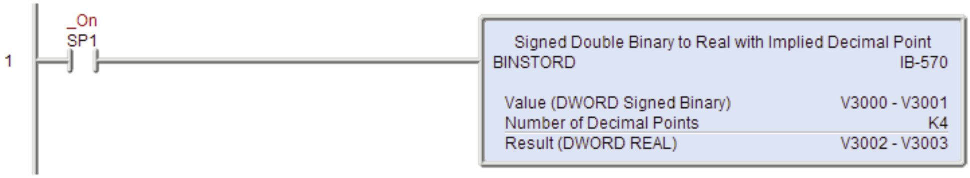
- Value (DWORD Signed Binary): The first V-Memory location where the 32-bit Signed Binary (decimal) value is located, or the constant value to convert.
- Number of Decimal Points: The desired number of digits to the right of the decimal point in the result.
- Result (DWORD Real): The first V-Memory location where the 32-bit Real (floating point) result will be stored.

Parameter	DL405 Range
Value . . . . V,K,P	K0 - 2147483647, All P Memory, All User V Memory
Number of Decimal Points. . . . . K	K0 - 10
Result V	All User V Memory

BINSTORD Example

In the following example the BINSTORD instruction is used to convert the signed double word binary value stored in V3000-V3001 to a 32 bit real number which is stored in V3002-V3003.

K4 in the decimal points implies that the data will have four digits to the right of the decimal point.



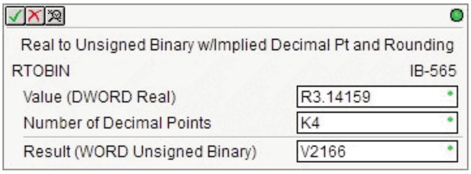
Signed Double Binary to Real with Implied Decimal Point
BINSTORD IB-570

Value (DWORD Signed Binary) V3000 - V3001
Number of Decimal Points K4
Result (DWORD REAL) V3002 - V3003

Real to Unsigned Binary with Implied Decimal Point and Rounding (RTOBIN) (IB-565)

DS6 **ONLY**	Used
HPP	N/A

The Real to Unsigned Binary with Implied Decimal Point and Rounding IBox converts the 32-bit real number to a 16-bit Unsigned Binary (decimal) value, compensating for an implied number of decimal points, then rounding the number up if needed.


Real to Unsigned Binary w/Implied Decimal Pt and Rounding
RTOBIN IB-565
Value (DWORD Real) R3.14159
Number of Decimal Points K4
Result (WORD Unsigned Binary) V2166

Example: R56.78 with the Number of Decimal Points set to K1 would yield the Binary value 568. If the Number of decimal Points is set to K0, this IBox would yield the Binary value 57 (the 6 is rounded up).

RTOBIN Parameters

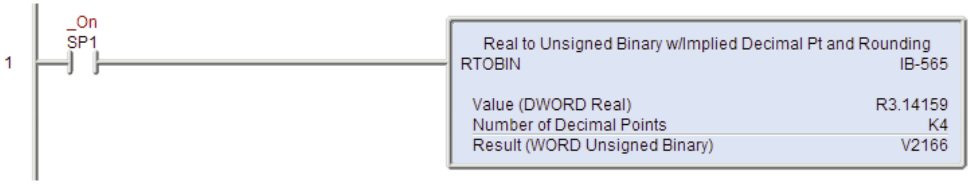
- Value (DWORD Real): The first V-Memory location where the 32-bit Real (floating point) value is located, or the constant value to convert.
- Number of Decimal Points: The desired number of digits to the right of the decimal point to convert.
- Result (WORD Unsigned Binary): The V-Memory location where the 16-bit Unsigned Binary (decimal) result will be stored.

Parameter	DL405 Range
Value . . . . V,R,P	R-3.402823E+38 - +3.402823E+38; All P Memory, All User V Memory
Number of Decimal Points. . . . . K	K0 - 5
Result V	All User V Memory

RTOBIN Example

In the following example the RTOBIN instruction is used to convert the real value R3.14159 to a binary number which is stored in V2166.

K4 in the decimal points implies that the data will have four digits to the right of the decimal point. The resulting value in V2166 is 31416.



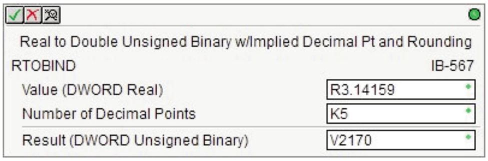
Real to Unsigned Binary w/Implied Decimal Pt and Rounding
RTOBIN IB-565

Value (DWORD Real) R3.14159
Number of Decimal Points K4
Result (WORD Unsigned Binary) V2166

Real to Double Unsigned Binary with Implied Decimal Point and Rounding (RTOBIND) (IB-567)

DS6 **ONLY**	Used
HPP	N/A

The Real to Double Unsigned Binary with Implied Decimal Point and Rounding IBox converts the 32-bit real number to a 32-bit Unsigned Binary (decimal) value, compensating for an implied number of decimal points, then rounding the number up if needed.



Real to Double Unsigned Binary w/Implied Decimal Pt and Rounding

RTOBIND IB-567

Value (DWORD Real) R3.14159

Number of Decimal Points K5

Result (DWORD Unsigned Binary) V2170

Example: R123456.78 with the Number of Decimal Points set to K2 would yield the BCD value 12345678. If the Number of decimal Points is set to K0, this IBox would yield the BCD value 123457 (the 6 is rounded up).

RTOBIND Parameters

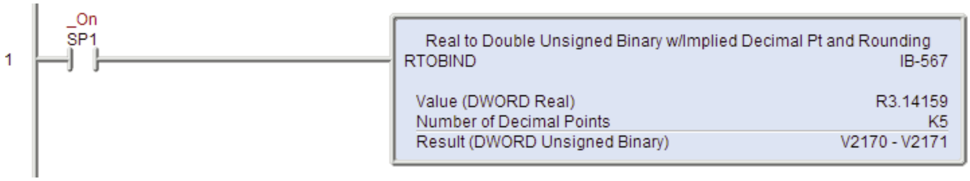
- Value (DWORD Real): The first V-Memory location where the 32-bit Real (floating point) value is located, or the constant value to convert.
- Number of Decimal Points: The desired number of digits to the right of the decimal point in the result.
- Result (DWORD Unsigned Binary): The first V-Memory location where the 32-bit Unsigned Binary (decimal) result will be stored.

Parameter	DL405 Range
Value . . . . V,R,P	R-3.402823E+38 - +3.402823E+38; All P Memory, All User V Memory
Number of Decimal Points. . . . . K	K0 - 10
Result V	All User V Memory

RTOBIND Example

In the following example the RTOBIND instruction is used to convert the real value R3.14159 to a double word binary number which is stored in V2170-V2071.

K5 in the decimal points implies that the data will have five digits to the right of the decimal point.The resulting value in V2170-V2171 is 314159.



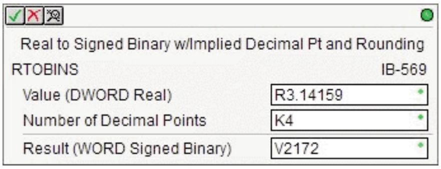
Real to Double Unsigned Binary w/Implied Decimal Pt and Rounding
RTOBIND IB-567

Value (DWORD Real)	R3.14159
Number of Decimal Points	K5
Result (DWORD Unsigned Binary)	V2170 - V2171

Real to Signed Binary with Implied Decimal Point and Rounding (RTOBINS) (IB-569)

DS6 **ONLY**	Used
HPP	N/A

The Real to Signed Binary with Implied Decimal Point and Rounding IBox converts the 32-bit real number to a 16-bit Signed Binary (decimal) value, compensating for an implied number of decimal points, then rounding the number up if needed.


Real to Signed Binary w/Implied Decimal Pt and Rounding
RTOBINS IB-569
Value (DWORD Real) R3.14159
Number of Decimal Points K4
Result (WORD Signed Binary) V2172

Example: R56.78 with the Number of Decimal Points set to K1 would yield the Binary value 568. If the Number of decimal Points is set to K0, this IBox would yield the Binary value 57 (the 6 is rounded up).

RTOBINS Parameters

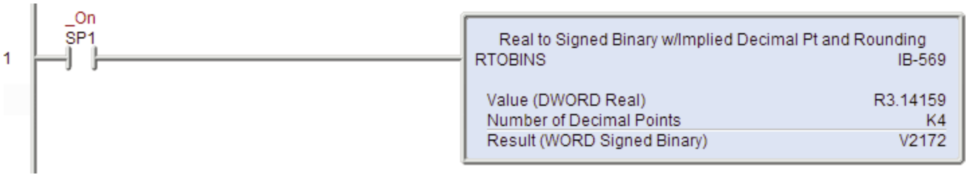
- Value (DWORD Real): The first V-Memory location where the 32-bit Real (floating point) value is located, or the constant value to convert.
- Number of Decimal Points: The desired number of digits to the right of the decimal point in the result.
- Result (WORD Signed Binary): The V-Memory location where the 16-bit Signed Binary (decimal) result will be stored.

Parameter	DL405 Range
Value . . . . V,R,P	R-3.402823E+38 - +3.402823E+38; All P Memory, All User V Memory
Number of Decimal Points. . . . . K	K0 - 5
Result V	All User V Memory

RTOBINS Example

In the following example the RTOBINS instruction is used to convert the real value R3.14159 to a signed binary number which is stored in V2172.

K4 in the decimal points implies that the data will have four digits to the right of the decimal point.The resulting value in V2172 is 31416.



Real to Signed Binary w/Implied Decimal Pt and Rounding

RTOBINS IB-569

Value (DWORD Real) R3.14159

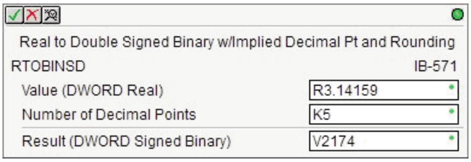
Number of Decimal Points K4

Result (WORD Signed Binary) V2172

Real to Double Signed Binary with Implied Decimal Point and Rounding (RTOBINSD) (IB-571)

DS6 **ONLY**	Used
HPP	N/A

The Real to Double Signed Binary with Implied Decimal Point and Rounding IBox converts the 32-bit real number to a 32-bit Signed Binary (decimal) value, compensating for an implied number of decimal points, then rounding the number up if needed.


Real to Double Signed Binary w/Implied Decimal Pt and Rounding
RTOBINSD IB-571
Value (DWORD Real) R3.14159
Number of Decimal Points K5
Result (DWORD Signed Binary) V2174

Example: R123456.78 with the Number of Decimal Points set to K2 would yield the value 12345678. If the Number of decimal Points is set to K0, this IBox would yield the value 123457 (the 6 is rounded up).

RTOBINSD Parameters

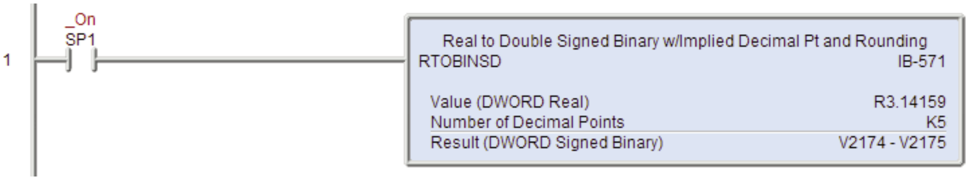
- Value (DWORD Real): The first V-Memory location where the 32-bit Real (floating point) value is located, or the constant value to convert.
- Number of Decimal Points: The desired number of digits to the right of the decimal point in the result.
- Result (DWORD Signed Binary): The first V-Memory location where the 32-bit Signed Binary (decimal) result will be stored.

Parameter	DL405 Range
Value . . . . V,R,P	R-3.402823E+38 - +3.402823E+38; All P Memory, All User V Memory
Number of Decimal Points. . . . . K	K0 - 10
Result V	All User V Memory

RTOBINSD Example

In the following example the RTOBINSD instruction is used to convert the real value R3.14159 to a signed double word binary number which is stored in V2174-V2175.

K5 in the decimal points implies that the data will have zero digits to the right of the decimal point.The resulting value in V2174-V2175 is 314159.

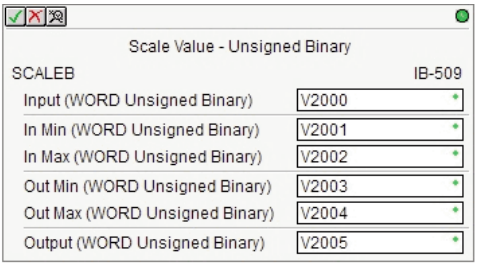


Real to Double Signed Binary w/Implied Decimal Pt and Rounding
RTOBINSD IB-571

Value (DWORD Real)	R3.14159
Number of Decimal Points	K5
Result (DWORD Signed Binary)	V2174 - V2175

Scale Value - Unsigned Binary (SCALEB) (IB-509)

DS6 ONLY	Used
HPP	N/A



Scale Value - Unsigned Binary

SCALEB IB-509

Input (WORD Unsigned Binary) V2000

In Min (WORD Unsigned Binary) V2001

In Max (WORD Unsigned Binary) V2002

Out Min (WORD Unsigned Binary) V2003

Out Max (WORD Unsigned Binary) V2004

Output (WORD Unsigned Binary) V2005

The Scale Value Unsigned Binary IBox will scale an unsigned 16-bit Binary value (0-65535) of a particular range into an unsigned 16-bit Binary value of another particular range.

This IBox only works with unsigned binary values, it DOES NOT work with signed binary or "sign plus magnitude" values.

The formula used is:

$$Output = \frac{(Input - InMin) \text{ x } (OutMax - OutMin)}{InMax - InMin} + OutMin$$

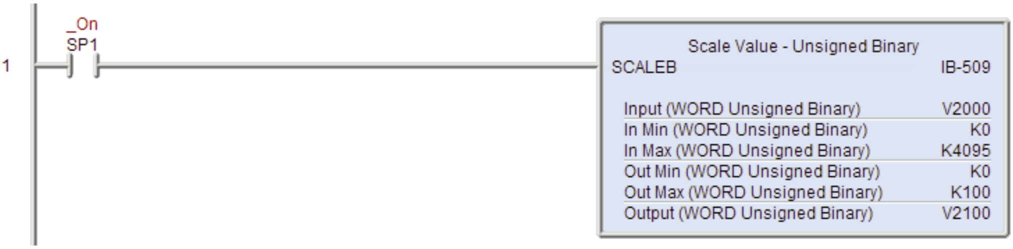
SCALEB Parameters

- Input (WORD Unsigned Binary): The raw 16-bit Unsigned Binary value to be scaled.
- In Min (WORD Unsigned Binary): The low limit (0-65535) of the Input range.
- In Max (WORD Unsigned Binary): The high limit (0-65535) of the Input range.
- Out Min (WORD Unsigned Binary): The low limit (0-65535) of the Output range.
- Out Max (WORD Unsigned Binary): The high limit (0-65535) of the Output range.
- Output (WORD Unsigned Binary): The scaled unsigned 16-bit Binary value (0-65535).

Parameter	DL405 Range
Input V	All V Memory
In Min . . . . V,K	K0 - 65535, All V Memory
In Max . . . . V,K	K0 - 65535, All V Memory
Out Min . . . . V,K	K0 - 65535, All V Memory
Out Max . . . . V,K	K0 - 65535, All V Memory
Output V	All User V Memory

SCALEB Example

In this SCALEB example a single word unsigned binary value from a 12 bit analog card in V2000 is being scaled from the 0 – 4095 raw value to 0 – 100 engineering units and the result is being stored in V2100 as a single word unsigned binary value. For example, if V2000 has a value of 2048 then the resulting value stored in V2100 is 50.



Scale Value - Unsigned Binary

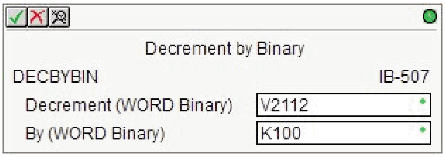
SCALEB IB-509

Input (WORD Unsigned Binary)	V2000
In Min (WORD Unsigned Binary)	K0
In Max (WORD Unsigned Binary)	K4095
Out Min (WORD Unsigned Binary)	K0
Out Max (WORD Unsigned Binary)	K100
Output (WORD Unsigned Binary)	V2100

Decrement By Binary (DECBYBIN) (IB-507)

DS6 **ONLY**	Used
HPP	N/A

The Decrement By Binary IBox will subtract the By (WORD Binary) Value from the Decrement (WORD Binary) Value on each scan the instruction is enabled.



Decrement by Binary

DECBYBIN IB-507

Decrement (WORD Binary) V2112

By (WORD Binary) K100

DECBYBIN Parameters

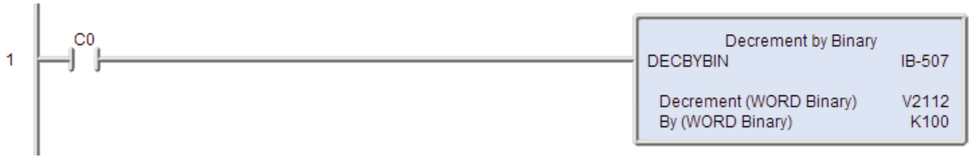
- Decrement (WORD Binary): The V-Memory location where the 16-bit Binary (decimal) value is located.
- By (WORD Binary): The WORD Binary (decimal) value to subtract.

Parameter	DL405 Range
Decrement V	All V Memory
By . . . . V,K	K0 - 65535, All V Memory

Discrete Bit Flags	Description
SP63	On when the result of the instruction causes the value in the accumulator to be zero.
SP64	On when the 16- bit subtraction instruction results in a borrow
SP65	On when the 32-bit subtraction instruction results in a borrow
SP70	On anytime the value in the accumulator is negative.

DECBYBIN Example

In this example the DECBYBIN instruction will subtract the value K100 from the binary value in V2112 on every scan that C0 is ON.

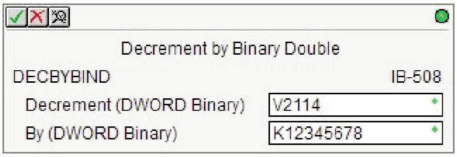


```
        Decrement by Binary
DECBYBIN                      IB-507

  Decrement (WORD Binary)      V2112
  By (WORD Binary)              K100
```

Decrement By Binary Double (DECBYBIND) (IB-508)

DS6 ONLY	Used
HPP	N/A

The Decrement By Binary Double IBox will subtract the By (DWORD Binary) Value from the Decrement (DWORD Binary) Value on each scan the instruction is enabled.



Decrement by Binary Double

DECBYBIND IB-508

Decrement (DWORD Binary): V2114

By (DWORD Binary): K12345678

DECBYBIND Parameters

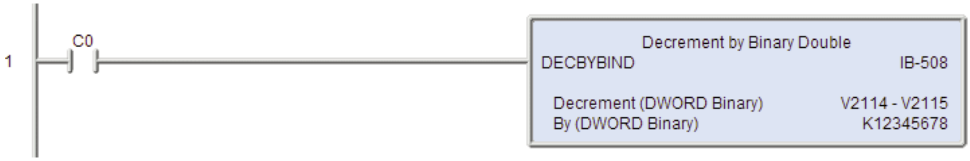
- Decrement (DWORD Binary): The V-Memory location where the 32-bit Binary Double (decimal) value is located.
- By (DWORD Binary): The DWORD Binary (decimal) value to subtract.

Parameter	DL405 Range
Decrement V	All V Memory
By . . . . V,K	K0 - 4294967295, All V Memory

Discrete Bit Flags	Description
SP63	On when the result of the instruction causes the value in the accumulator to be zero.
SP64	On when the 16- bit subtraction instruction results in a borrow
SP65	On when the 32-bit subtraction instruction results in a borrow
SP70	On anytime the value in the accumulator is negative.

DECBYBIND Example

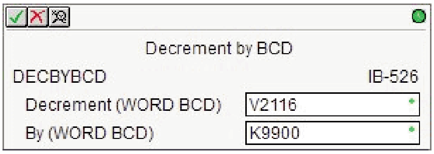
In this example the DECBYBIND instruction will subtract the value K12345678 from the double word binary value in V2114-V2115 on every scan that C0 is ON.



Decrement By BCD (DECBYBCD) (IB-526)

DS6 **ONLY**	Used
HPP	N/A

The Decrement By BCD IBox will subtract the By (WORD BCD) Value from the Decrement (WORD BCD) Value on each scan the instruction is enabled.



Decrement by BCD

DECBYBCD IB-526

Decrement (WORD BCD) V2116

By (WORD BCD) K9900

DECBYBCD Parameters

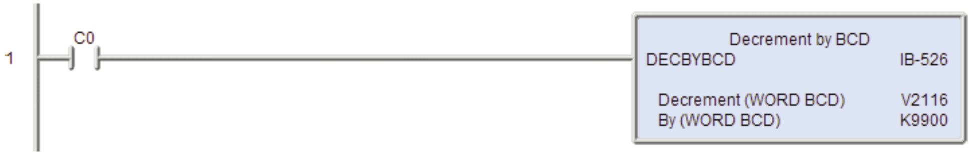
- Decrement (WORD BCD): The V-Memory location where the 16-bit BCD value is located.
- By (WORD BCD): The WORD BCD value to subtract.

Parameter	DL405 Range
Decrement V	All V Memory
By . . . . V,K	K0 - 9999, All V Memory

Discrete Bit Flags	Description
SP63	On when the result of the instruction causes the value in the accumulator to be zero.
SP64	On when the 16- bit subtraction instruction results in a borrow
SP65	On when the 32-bit subtraction instruction results in a borrow
SP70	On anytime the value in the accumulator is negative.
SP75	On when a BCD instruction is executed and a NON–BCD number was encountered.

DECBYBCD Example

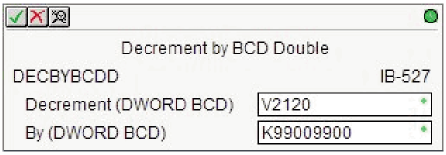
In this example the DECBYBCD instruction will subtract the BCD value K9900 from the BCD value in V2116 on every scan that C0 is ON.



Decrement By BCD Double (DECBYBCDD) (IB-527)

DS6 **ONLY**	Used
HPP	N/A

The Decrement By BCD Double IBox will subtract the By (DWORD BCD) Value from the Decrement (DWORD BCD) Value on each scan the instruction is enabled.



Decrement by BCD Double

DECBYBCDD IB-527

Decrement (DWORD BCD) V2120

By (DWORD BCD) K99009900

DECBYBCDD Parameters

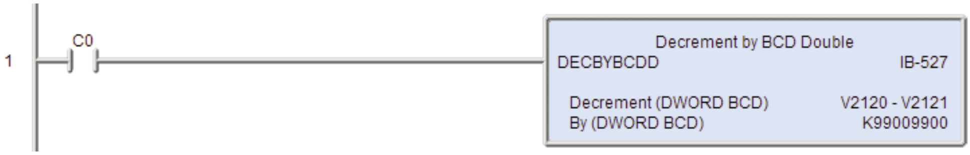
- Decrement (DWORD BCD): The V-Memory location where the 32-bit BCD value is located.
- By (DWORD BCD): The DWORD BCD value to subtract.

Parameter	DL405 Range
Decrement V	All V Memory
By . . . . V,K	K0 - 99999999, All V Memory

Discrete Bit Flags	Description
SP63	On when the result of the instruction causes the value in the accumulator to be zero.
SP64	On when the 16- bit subtraction instruction results in a borrow
SP65	On when the 32-bit subtraction instruction results in a borrow
SP70	On anytime the value in the accumulator is negative.
SP75	On when a BCD instruction is executed and a NON–BCD number was encountered.

DECBYBCDD Example

In this example the DECBYBCDD instruction will subtract the BCD value K99009900 from the double word BCD value in V2120-V2121 on every scan that C0 is ON.

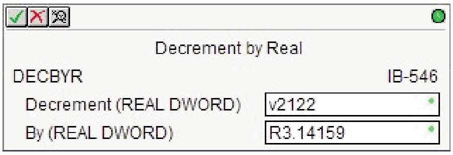


Decrement (DWORD BCD) V2120 - V2121
By (DWORD BCD) K99009900

Decrement By Real (DECBYR) (IB-546)

DS6 ONLY	Used
HPP	N/A

The Decrement By Real IBox will subtract the By (REAL DWORD) Value from the Decrement (REAL DWORD) Value on each scan the instruction is enabled.



Decrement by Real

DECBYR IB-546

Decrement (REAL DWORD) v2122

By (REAL DWORD) R3.14159

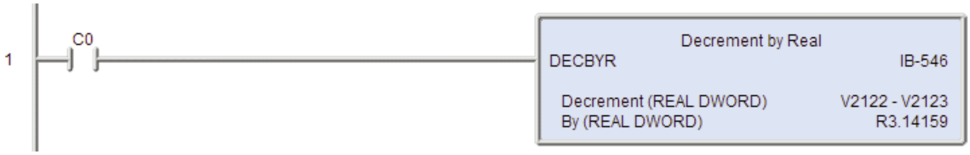
DECBYR Parameters

- Decrement (Real DWORD): The first V-Memory location where the 32-bit Real (floating point) value is located.
- By (Real DWORD): The 32-bit Real (floating point) value to subtract.

Parameter	DL405 Range
Decrement V	All V Memory
By . . . . V,R	R-3.402823E+38 - +3.402823E+38, All V Memory

DECBYR Example

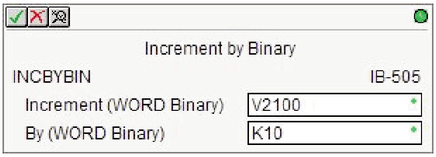
In this example the DECBYR instruction will subtract the real value R3.14159 from the real value in V2122-V2123 on every scan that C0 is ON.



Increment By Binary (INCBYBIN) (IB-505)

DS6 ONLY	Used
HPP	N/A

The Increment By Binary IBox will add the By (WORD Binary) Value to the Increment (WORD Binary) Value on each scan the instruction is enabled.



Increment by Binary

INCBYBIN IB-505

Increment (WORD Binary): V2100

By (WORD Binary): K10

INCBYBIN Parameters

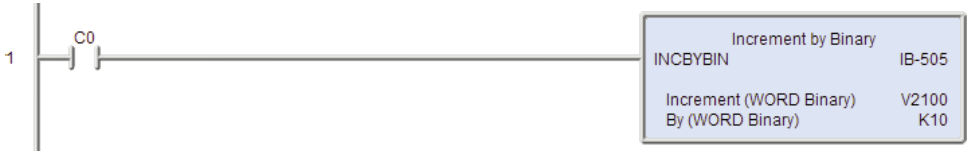
- Increment (WORD Binary): The V-Memory location where the 16-bit Binary (decimal) value is located.
- By (WORD Binary): The WORD Binary (decimal) value to add.

Parameter	DL405 Range
Increment V	All V Memory
By . . . . V,K	K0 - 65535, All V Memory

Discrete Bit Flags	Description
SP63	On when the result of the instruction causes the value in the accumulator to be zero.
SP66	On when the 16-bit addition instruction results in a carry.
SP67	On when the 32-bit addition instruction results in a carry.
SP70	On anytime the value in the accumulator is negative.
SP73	On when a signed addition or subtraction results in an incorrect sign bit.

INCBYBIN Example

In this example the INCBYBIN instruction will add the value K10 to the binary value in V2100 on every scan that C0 is ON.



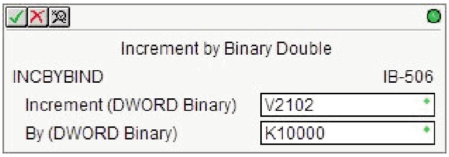
Increment by Binary
INCBYBIN IB-505

Increment (WORD Binary) V2100
By (WORD Binary) K10

Increment By Binary Double (INCBYBIND) (IB-506)

DS6 **ONLY**	Used
HPP	N/A

The Increment By Binary Double IBox will add the By (DWORD Binary) Value to the Increment (DWORD Binary) Value on each scan the instruction is enabled.



Increment by Binary Double

INCBYBIND IB-506

Increment (DWORD Binary) V2102

By (DWORD Binary) K10000

INCBYBIND Parameters

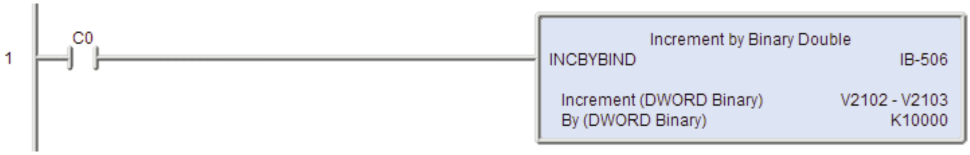
- Increment (DWORD Binary): The V-Memory location where the 32-bit Binary Double (decimal) value is located.
- By (DWORD Binary): The DWORD Binary (decimal) value to add.

Parameter	DL405 Range
Increment V	All V Memory
By . . . . V,K	K0 - 4294967295, All V Memory

Discrete Bit Flags	Description
SP63	On when the result of the instruction causes the value in the accumulator to be zero.
SP66	On when the 16-bit addition instruction results in a carry.
SP67	On when the 32-bit addition instruction results in a carry.
SP70	On anytime the value in the accumulator is negative.
SP73	On when a signed addition or subtraction results in an incorrect sign bit.

INCBYBIND Example

In this example the INCBYBIND instruction will add the value K10000 to the double word binary value in V2102-V2103 on every scan that C0 is ON.

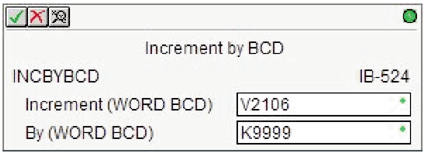


Increment (DWORD Binary) V2102 - V2103
By (DWORD Binary) K10000

Increment By BCD (INCBYBCD) (IB-524)

DS6 **ONLY**	Used
HPP	N/A

The Increment By BCD IBox will add the By (WORD BCD) Value to the Increment (WORD BCD) Value on each scan the instruction is enabled.



Increment by BCD

INCBYBCD IB-524

Increment (WORD BCD) V2106

By (WORD BCD) K9999

INCBYBCD Parameters

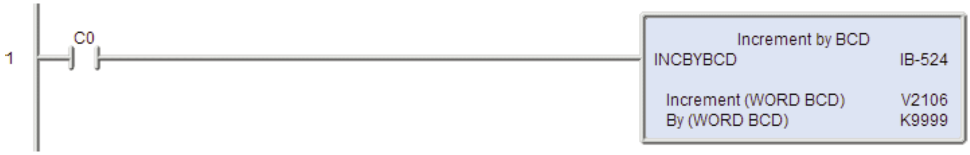
- Increment (WORD BCD): The V-Memory location where the 16-bit BCD value is located.
- By (WORD BCD): The WORD BCD value to add.

Parameter	DL405 Range
Increment V	All V Memory
By . . . . V,K	K0 - 9999, All V Memory

Discrete Bit Flags	Description
SP63	On when the result of the instruction causes the value in the accumulator to be zero.
SP66	On when the 16-bit addition instruction results in a carry.
SP67	On when the 32-bit addition instruction results in a carry.
SP70	On anytime the value in the accumulator is negative.
SP73	On when a signed addition or subtraction results in an incorrect sign bit.
SP75	On when a BCD instruction is executed and a NON–BCD number was encountered.

INCBYBCD Example

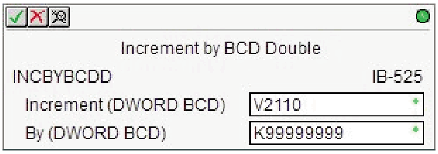
In this example the INCBYBCD instruction will add the BCD value K9999 to the binary value in V2106 on every scan that C0 is ON.



Increment By BCD Double (INCBYBCDD) (IB-525)

DS6 **ONLY**	Used
HPP	N/A

The Increment By BCD Double IBox will add the By (DWORD BCD) Value to the Increment (DWORD BCD) Value on each scan the instruction is enabled.


Increment by BCD Double

INCBYBCDD IB-525

Increment (DWORD BCD) V2110

By (DWORD BCD) K99999999

INCBYBCDD Parameters

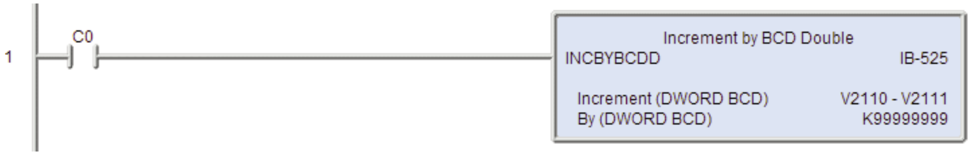
- Increment (DWORD BCD): The V-Memory location where the 32-bit BCD value is located.
- By (DWORD BCD): The DWORD BCD value to add.

Parameter	DL405 Range
Increment V	All V Memory
By . . . . V,K	K0 - 99999999, All V Memory

Discrete Bit Flags	Description
SP63	On when the result of the instruction causes the value in the accumulator to be zero.
SP66	On when the 16-bit addition instruction results in a carry.
SP67	On when the 32-bit addition instruction results in a carry.
SP70	On anytime the value in the accumulator is negative.
SP73	On when a signed addition or subtraction results in an incorrect sign bit.
SP75	On when a BCD instruction is executed and a NON–BCD number was encountered.

INCBYBCDD Example

In this example the INCBYBCDD instruction will add the BCD value K99999999 to the BCD value in V2110-V2111 on every scan that C0 is ON.



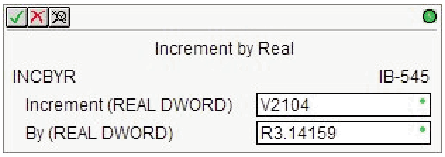
Increment by BCD Double
INCBYBCDD IB-525

Increment (DWORD BCD) V2110 - V2111
By (DWORD BCD) K99999999

Increment By Real (INCBYR) (IB-545)

DS6 ONLY	Used
HPP	N/A

The Increment By Real IBox will add the By (REAL DWORD) Value to the Increment (REAL DWORD) Value on each scan the instruction is enabled.



Increment by Real

INCBYR IB-545

Increment (REAL DWORD) V2104

By (REAL DWORD) R3.14159

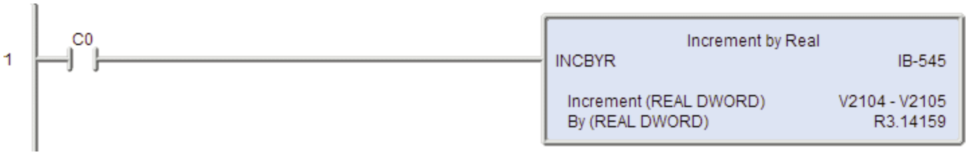
INCBYR Parameters

- Increment (Real DWORD): The first V-Memory location where the 32-bit Real (floating point) value is located.
- By (Real DWORD): The 32-bit Real (floating point) value to add.

Parameter	DL405 Range
Increment V	All V Memory
By . . . . V,R	R-3.402823E+38 - +3.402823E+38, All V Memory

INCBYR Example

In this example the INCBYR instruction will add the real value R3.14159 to the real value in V2104-V2105 on every scan that C0 is ON.



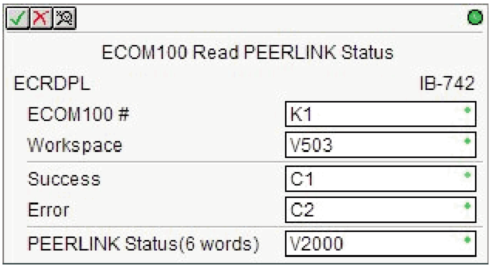
Increment by Real
INCBYR IB-545

Increment (REAL DWORD) V2104 - V2105
By (REAL DWORD) R3.14159

ECOM100 Read PEERLINK Status (ECRDPL) (IB-742)

DS6 ONLY	Used
HPP	N/A

The ECOM100 Read PEERLINK Status IBox will read the PEERLINK operation's runtime status information from an ECOM100 that is configured to be part of a PEERLINK network. This IBox will return 6 registers that contain information about current PEERLINK status and configuration.


ECOM100 Read PEERLINK Status
ECRDPL IB-742
ECOM100 # K1
Workspace V503
Success C1
Error C2
PEERLINK Status(6 words) V2000

It references the ECOM100 # of the ECOM100 Config IBox that is controlling the ECOM100 module in a specific slot. The ECOM100 Config contains built-in interlocking logic that is used to synchronize the processing of this IBox with all of the other IBoxes in the ladder program that are being processed by the same ECOM100.

A PEERLINK network is a data sharing network that consists of any number of DirectLOGIC PLC and/or Do-more PLC systems using ECOM100 modules and/or the Do-more PLC's onboard Ethernet port. Each member of the data sharing network can receive data from the other members on the data sharing network by "subscribing to" them, or send data to the other members of the network by electing to "publish" one or more blocks of PEERLINK memory.

When PEERLINK is configured in an ECOM100 the user specifies a section of V-Memory that is allocated for exclusive use by the PEERLINK operation. This memory contains 256 locations. These 256 locations are divided into 16 blocks. Each of these 16 data blocks consists of 16-Bit registers. Theses blocks provide the local storage for the data that is sent and received over the data-sharing network.

PEERLINK uses the verbs 'publishing' and 'subscribing' to describe how data is exchanged with ECOM100s on the data sharing network. Publishing is analogous to sending data, and is done only if the PEERLINK configuration is set to 'publish' one or more of its own data blocks. If so configured, the ECOM100 will broadcast a packet that contains all of the data from the V-Memory blocks. There are sixteen unique data blocks, and each data block can only be published by one ECOM100 or Do-more PLC. This means there can be a maximum of sixteen unique ECOMs configured to publish blocks of data. A single ECOM100 can be configured so that it publishes none of the blocks, one block, some of the blocks, or even all 16 of the blocks.

Subscribing is analogous to receiving data, and is accomplished by 'subscribing to' the data blocks of all the other controllers on the data sharing network. Once PEERLINK is enabled, it listens to the network for PEERLINK broadcasts messages from other ECOM100s or Do-more PLCs. When it receives one, it examines the data from that packet, and for blocks that are configured as "Subscribe To", it stores that data in the controller's local V-Memory in the appropriate block.

The PEERLINK network uses TCP/IP broadcast packets to publish the blocks of data to the network. One caveat with the use of broadcast packets is that it limits the scope of the shared data network to the local broadcast domain.

The ECOM100 Read PEERLINK Status IBox retrieves 6 status values from the ECOM100 and places those values in 6 consecutive V-Memory locations. The definitions of those 6 status values follows:

Number	Name	Description
Word 1	Paused	1 = PEERLINK processing is Paused in this ECOM100 0 = PEERLINK processing is Active
Word 2	PEERLINK Enabled	1 = PEERLINK is Enabled in this ECOM100 0 = PEERLINK is NOT Enabled in this ECOM100
Word 3	PEERLINK Address	The first of the 256 V-Memory locations that the PEERLINK operation uses for storing the data that is sent and received through the Publish and Subscribe operations
Word 4	Ignored Blocks	Indicates which of the 16 PEERLINK blocks are being ignored by this ECOM100. If the bit is ON the block is being ignored, if the bit is OFF the block is NOT ignored. Each of the 16 bits in this Word corresponds to a PEERLINK block as follows: Bit 0 = Block 0 Bit 1 = Block 1 ... Bit 14 = Block 14 Bit 15 = Block 15
Word 5	Published Blocks	Indicates which of the 16 PEERLINK blocks are being published by this ECOM100. If the bit is ON the block is being published, if the bit is OFF the block is NOT being published. Each of the 16 bits in this Word corresponds to a PEERLINK block as follows: Bit 0 = Block 0 Bit 1 = Block 1 ... Bit 14 = Block 14 Bit 15 = Block 15
Word 6	Subscribed Blocks	Indicates which of the 16 PEERLINK blocks this ECOM100 is subscribing to. If the bit is ON the block is being subscribed to, if the bit is OFF the block is NOT being subscribed to. Each of the 16 bits in this Word corresponds to a PEERLINK block as follows: Bit 0 = Block 0 Bit 1 = Block 1 ... Bit 14 = Block 14 Bit 15 = Block 15

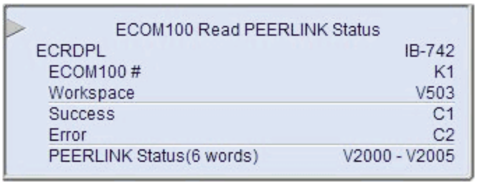
ECRDPL Parameters

- ECOM100#: This is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number.
- Workspace: A V-Memory register that is used internally by this IBox. It must not be used by any other instructions in the PLC.
- Success: This BIT will be ON if the ECRDPL succeeds and OFF if the ECRDPL fails.
- Error: This BIT will be OFF if the ECRDPL succeeds and ON if the ECRDPL fails.
- PEERLINK Status (6 Words): The first of the 6 consecutive V-Memory registers where the PEERLINK Status values will be stored.

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	All User V Memory
Success . . . . X,Y,C,GX,GY,B	All Bit Memory
Error . . . . X,Y,C,GX,GY,B	All Bit Memory
PEERLINK Status V	All User V Memory

***Note:** When the ECRDPL IBox is allowed to execute, the Success and Error BITs are both set to OFF. One of these Bits is guaranteed to be ON after the IBox execution is complete. These BITs will retain their ON/OFF value until the IBox is executed again.*

***Note:** The gray triangle at the right end of an input leg indicates the input is edge triggered. Meaning that each time the input logic transitions from OFF to ON this instruction will execute.*

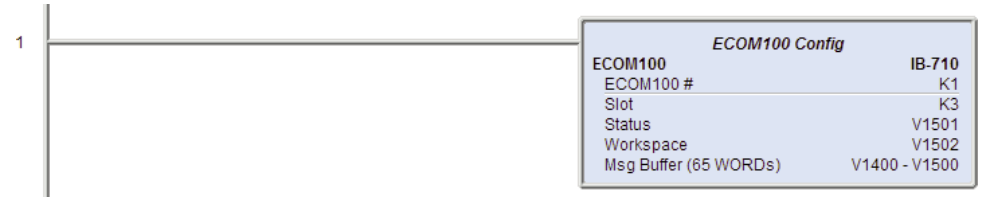


ECOM100 Read PEERLINK Status	
ECRDPL	IB-742
ECOM100 #	K1
Workspace	V503
Success	C1
Error	C2
PEERLINK Status(6 words)	V2000 - V2005

With each execution, this instruction will run to completion even if the input logic transitions to OFF before the instruction completes.

ECRDPL Example

Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 3 as ECOM100# K1. All other ECxxxx IBoxes refer to this module # as K1. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V1501 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module. V1502 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V1400-V1500 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



ECOM100 Config

ECOM100 **IB-710**

ECOM100 # K1

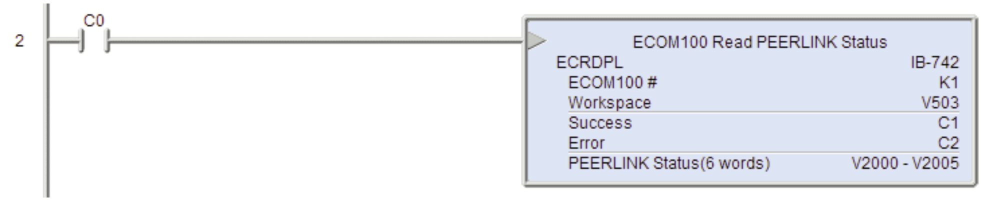
Slot K3

Status V1501

Workspace V1502

Msg Buffer (65 WORDs) V1400 - V1500

Rung 2: Each time that C0 is enabled, 6 PEERLINK status locations will be read from the ECOM100 and stored in V2000-V2005. C1 will be enabled if the read is a success, C2 will be enabled if the attempted read results in failure.



ECOM100 Read PEERLINK Status

ECRDPL IB-742

ECOM100 # K1

Workspace V503

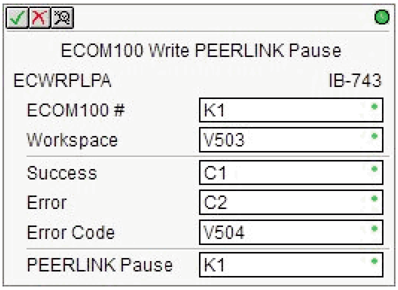
Success C1

Error C2

PEERLINK Status(6 words) V2000 - V2005

ECOM100 Write PEERLINK Pause (ECWRPLPA) (IB-743)

DS6 ONLY	Used
HPP	N/A



ECOM100 Write PEERLINK Pause

ECWRPLPA IB-743

ECOM100 # K1

Workspace V503

Success C1

Error C2

Error Code V504

PEERLINK Pause K1

The ECOM100 Write PEERLINK Pause IBox will Enable and/or Disable the PEERLINK processing in the specified ECOM100.

It references the ECOM100 # of the ECOM100 Config IBox that is controlling the ECOM100 module in a specific slot. The ECOM100 Config contains built-in interlocking logic that is used to synchronize the processing of this IBox with all of the other IBoxes in the ladder program that are being processed by the same ECOM100.

ECWRPLPA Parameters

- ECOM100#: This is a logical number associated with this specific ECOM100 module in the specified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number.
- Workspace: A V-Memory register that is used internally by this IBox. It must not be used by any other instructions in the PLC.
- Success: This BIT will be ON if the Write operation succeeds and OFF if the Write operation fails.
- Error: This BIT will be OFF if the Write operation succeeds and ON if the Write operation fails.
- Error Code: A V-Memory register that stores the Return Code from the ECOM100 if the Write operation fails. It must not be used by any other instructions in the PLC.

  The possible Error Return Codes are:

  0 = No Error

  126 = Write Protect Error - the ECOM100 is configured to use DIP Switch 5 to write protect the ECOM100, and DIP 5 is ON
- PEERLINK Pause: The value to write, either a constant or a V-Memory location that contains the following values:

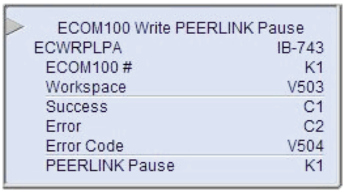
  0 = Allow PEERLINK operation

  1 = Pause PEERLINK operation

Parameter	DL405 Range
ECOM100# . . . . K	K0-255
Workspace V	All User V Memory
Success . . . . X,Y,C,GX,GY,B	All Bit Memory
Error . . . . X,Y,C,GX,GY,B	All Bit Memory
Error Code . . . . X,Y,C,GX,GY,B	All Bit Memory
PEERLINK Pause . . . . V,K	K0-1, All User V Memory

***Note:** When the ECWRPLPA IBox is allowed to execute, the Success and Error BITs are both set to OFF. One of these Bits is guaranteed to be ON after the IBox execution is complete. These BITs will retain their ON/OFF value until the IBox is executed again.*

***Note:**The gray triangle at the right end of an input leg indicates the input is edge triggered. Meaning that each time the input logic transitions from OFF to ON this instruction will execute.*



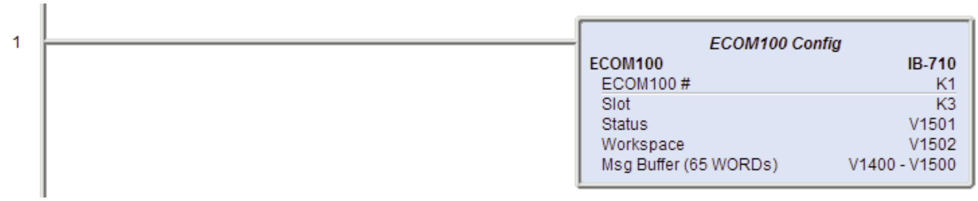
ECOM100 Write PEERLINK Pause

ECWRPLPA	IB-743
ECOM100 #	K1
Workspace	V503
Success	C1
Error	C2
Error Code	V504
PEERLINK Pause	K1

With each execution, this instruction will run to completion even if the input logic transitions to OFF before the instruction completes.

ECWRPLPA Example

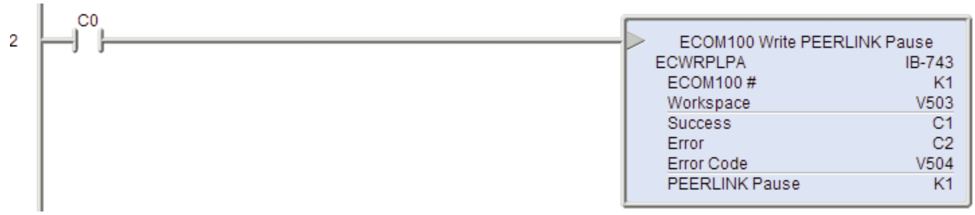
Rung 1: The ECOM100 Config IBox is responsible for coordination/interlocking of all ECOM100 type IBoxes for one specific ECOM100 module. Tag the ECOM100 in slot 3 as ECOM100# K1. All other ECxxxx IBoxes refer to this module # as K1. If you need to move the module in the base to a different slot, then you only need to change this one IBox. V1501 is used as a global result status register for the other ECxxxx IBoxes using this specific ECOM100 module. V1502 is used to coordinate/interlock the logic in all of the other ECxxxx IBoxes using this specific ECOM100 module. V1400-V1500 is a common 130 byte buffer available for use by the other ECxxxx IBoxes using this specific ECOM100 module.



ECOM100 Config

ECOM100	IB-710
ECOM100 #	K1
Slot	K3
Status	V1501
Workspace	V1502
Msg Buffer (65 WORDs)	V1400 - V1500

Rung 2: Each time that C0 is enabled, K1 will be sent to the ECOM100 module to pause the PEERLINK feature. A K0 would need to be sent to resume PEERLINK operation. C1 will be enabled if the pause is a success, C2 will be enabled if the attempted pause results in failure.



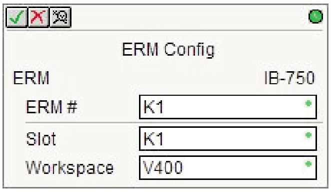
ECOM100 Write PEERLINK Pause

ECWRPLPA	IB-743
ECOM100 #	K1
Workspace	V503
Success	C1
Error	C2
Error Code	V504
PEERLINK Pause	K1

ERM Config (ERM) (IB-750)

DS6 ONLY	Used
HPP	N/A

The ERM Config IBox defines all of the information necessary to setup an ERM or ERM100 for use by other ERM-specific IBoxes (ERxxxxx). The ERM Config IBox is the resource manager for the slot or port it is setup to use. It will internally monitor the "Busy" and "Error" SP bits so that it can control all of the other ERM-specific IBoxes in the ladder program.



ERM Config

ERM IB-750

ERM # K1

Slot K1

Workspace V400

ERM Config IBox requirements:

- If you wish to use any of the ERM IBoxes, you must have an ERM Config IBox for each ERM and ERM100 module in the system.
- The ERM Config IBox must be located at the top of the ladder or stage program.
- The ERM Config IBox is "always ON", so it can not have any input logic. This IBox must be in a section of the ladder program that is always enabled, so do not place it in a Stage or a subroutine that will ever be disabled.
- The ERM-specific IBoxes require that DIP switch #7 be set ON.

ERM Parameters

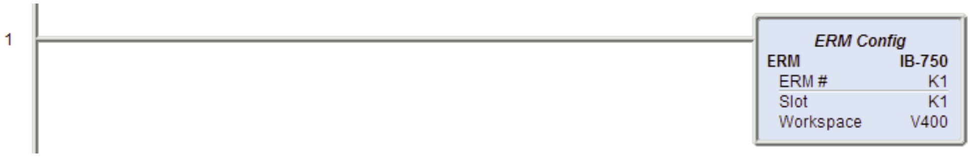
- ERM#: A reference number or resource number used to uniquely identify the ERM network.
- Slot: Identifies which slot contains the ERM or ERM100 module.
- Workspace: A V-Memory register that is used internally by this IBox. It must not be used by any other instructions in the PLC.

Parameter	DL405 Range
ERM# . . . . K	K0 - 255
Slot . . . . K	K0 - 7
Workspace V	All User V Memory

Note: *No input logic is allowed on the rung with this IBox.*

ERM Example

Rung 1: The ERM Config IBox is responsible for coordination/interlocking of all ERM type IBoxes for one specific ERM module. Tag the ERM in slot 1 as ERM# K1. All other ERxxxx IBoxes refer to this module # as K1. If you need to move the module in the base to a different slot, then you only need to change this one IBox.



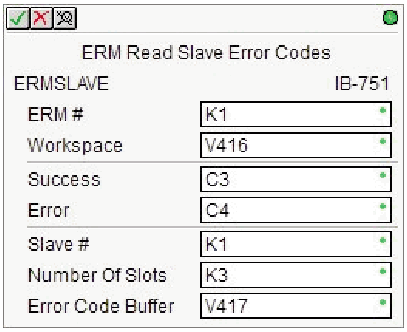
ERM # K1
Slot K1
Workspace V400

ERM Read Slave Error Codes (ERMSLAVE) (IB-751)

DS6 ONLY	Used
HPP	N/A

The ERM Read Slave Error Codes IBox will read the error information from a Slave that is part of an ERM network. Each Slave will return 4 Words of data plus 1 Word for each I/O slot in that slave's base. A maximum of 36 words of error code data can be read from a single slave depending on the number of bases and I/O modules (slots) used per slave.

The program will need a separate ERM Read Slave Error Codes for each slave on the ERM network.



ERM Read Slave Error Codes

ERMSLAVE IB-751

ERM # K1

Workspace V416

Success C3

Error C4

Slave # K1

Number Of Slots K3

Error Code Buffer V417

This IBox references the ERM # of the ERM Config IBox that is controlling the ERM or ERM100 module in a specific slot. The ERM Config contains built-in interlocking logic that is used to synchronize the processing of this IBox with all of the other IBoxes in the ladder program that are being processed by the same ERM or ERM100 module.

ERMSLAVE Parameters

- ERM#: A reference number or resource number used to uniquely identify the ERM network.
- Workspace: A V-Memory register that is used internally by this IBox. It must not be used by any other instructions in the PLC.
- Success: This BIT will be ON if the Read Error Code succeeds and OFF if the Read Error Code fails.
- Error: This BIT will be OFF if the Read Error Code succeeds and ON if the Read Error Code fails.
- Slave#: The number of the ERM Slave to Read the Error Codes from. This number is the order in which they appear in the ERM network configuration in ERM Workbench.
- Number of Slots: The number of Slots in the specified ERM slave.
- Error Code Buffer: The first of the consecutive V-Memory registers where the ERM Error code values will be stored. The status buffer consumes 4 V-Memory locations + one V-Memory location for each Slot as specified above.

Parameter	DL405 Range
ERM# . . . . K	K0-255
Workspace V	All User V Memory
Success . . . . X,Y,C,GX,GY,B	All Bit Memory
Error . . . . X,Y,C,GX,GY,B	All Bit Memory
Slave# . . . . K	K1-16
Number of Slots . . . . K	K1-32
Error Code Buffer V	All User V Memory

The Error Code Buffer stores error information in the following sequence:

Word Offset	Name	Description
0	Current Error Code	The current error code reported by the Slave Bits 0 - 11: Error Code Bit 12: ON = I/O Error Bit 13: ON = I/O Warning Bit 14: n/a Bit 15: n/a
1	Module Slot (0-15)	For slots 0 - 15, the I/O slot that has a module reporting an error.
2	Module Slot (16-31)	For slots 16 - 31, the I/O slot that has a module reporting an error.
3	Last Error Code	The previous error code reported by the Slave Bits 0 - 11: Error Code Bit 12: ON = I/O Error Bit 13: ON = I/O Warning Bit 14: n/a Bit 15: n/a
4	Ext Error Code Local Base Slot 0	
...	...	
11	Ext Error Code Local Base Slot 7	
12	Ext Error Code Slot 8 or Expansion Base 1 Slot 0	
...	...	
19	Ext Error Code Slot 15 or Expansion Base 1 Slot 7	
20	Ext Error Code Slot 16 Expansion Base 2 Slot 0	
...	...	
27	Ext Error Code Slot 23 or Expansion Base 2 Slot 7	
28	Ext Error Code Slot 24 or Expansion Base 3 Slot 0	
...	...	
35	Ext Error Code Slot 31 or Expansion Base 3 Slot 7	

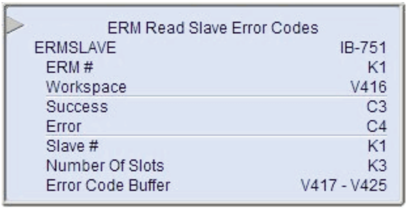
The following chart has the Slave Error Codes for Word 0 and Word 3 in previous table. These error codes are valid for DL205, DL405, and Terminator I/O slaves.

Code (decimal)	Description
0	No Error
121	Channel Failure
122	Unused Channels Exist - the module has jumpers to disable unused channels
139	Broken Transmitter on one of the analog input channels
142	Multiple Channel Failure
153	Terminator I/O Slave only (Hot-Swap Error): The module which was in this slot is no longer responding, most likely because the user has manually removed an I/O module. If Automatic Reset (default) is enabled for this slave, it will reset itself once the replacement module is inserted. If Manual Reset is enabled for this slave, the user must do the following: 1. SET the slave disable flag for that slave in the first diagnostic output word 2. Wait for bits 8 - 15 in second diagnostic input word to equal 1 3. RESET the slave disable flag in the first diagnostic output word.
154	Terminator I/O Slave only (Hot-Swap Error): I/O configuration has changed, most likely because the user has manually added an I/O module. See 153 above for reset methods.
155	Terminator I/O Slave only (Module Error): One or more of the I/O modules has an error. For more detail check extended errors.
200-216	Unused analog input channels exist at channel xx (1 - 16), where xx = Value - 200. For example: 212 indicates unused analog channel exists at channel 12.

The following chart has the Extended Slave Error Codes for Words 4 through 35 in the Error Code Buffer. These error codes are valid for DL205, DL405, and Terminator I/O slaves.

<table>
<tr><th>Code
(decimal)</th><th colspan="2">Description</th></tr>
<tr><td rowspan="9">32 - 63</td><td colspan="2">Bit-wise error where bit 5 is always SET.
Look at bit 0 thru bit 4 to get a possible list of errors.
For example: 34 decimal = 22 hexadecimal = 0010_0010 (Bit 5 and Bit 1 ON).</td></tr>
<tr><td>Bit Number</td><td>Description</td></tr>
<tr><td>0</td><td>Terminal block off</td></tr>
<tr><td>1</td><td>External P/S voltage low</td></tr>
<tr><td>2</td><td>Fuse blown</td></tr>
<tr><td>3</td><td>Bus error</td></tr>
<tr><td>4</td><td>Module initialization error (intelligent module)</td></tr>
<tr><td>5</td><td>Fault exists in module</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td>117</td><td colspan="2">Write attempt to an invalid analog channel.</td></tr>
<tr><td>119</td><td colspan="2">Data not valid. Subnet mask or IP address not allowed. Likely because the data packet is not constructed properly.</td></tr>
<tr><td>121</td><td colspan="2">Analog input channel error.</td></tr>
<tr><td>122</td><td colspan="2">Unused analog input channels exist.</td></tr>
<tr><td>139</td><td colspan="2">Broken Transmitter on one of the analog input channels</td></tr>
<tr><td>142</td><td colspan="2">Channel Failure</td></tr>
<tr><td>146</td><td colspan="2">Communications failure. HA-EDRV2 onboard relay has tripped</td></tr>
<tr><td>153</td><td colspan="2">Terminator I/O Slave only (Hot-Swap):
The module which was in this slot is no longer responding, most likely because the user has manually removed an I/O module.</td></tr>
<tr><td>154</td><td colspan="2">Terminator I/O Slave only (Hot-Swap):
I/O configuration has changed, most likely because the user has manually added an I/O module.</td></tr>
<tr><td>155</td><td colspan="2">Terminator I/O Slave only (Module Error):
One or more of the I/O modules has an error.</td></tr>
<tr><td>200-216</td><td colspan="2">Unused analog input channels exist at channel xx (1 - 16), where xx = Value - 200.
For example: 212 indicates unused analog channel exists at channel 12.</td></tr>
</table>

__Note:__The gray triangle at the right end of an input leg indicates the input is edge triggered. Meaning that each time the input logic transitions from OFF to ON this instruction will execute.



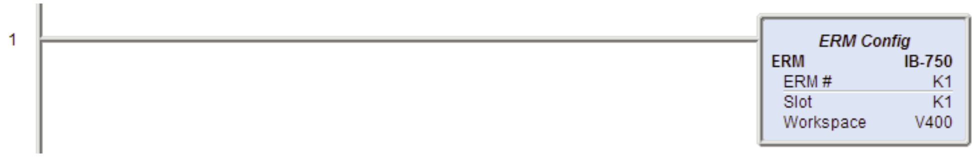
ERM Read Slave Error Codes

ERMSLAVE	IB-751
ERM #	K1
Workspace	V416
Success	C3
Error	C4
Slave #	K1
Number Of Slots	K3
Error Code Buffer	V417 - V425

With each execution, this instruction will run to completion even if the input logic transitions to OFF before the instruction completes.

ERMSLAVE Example

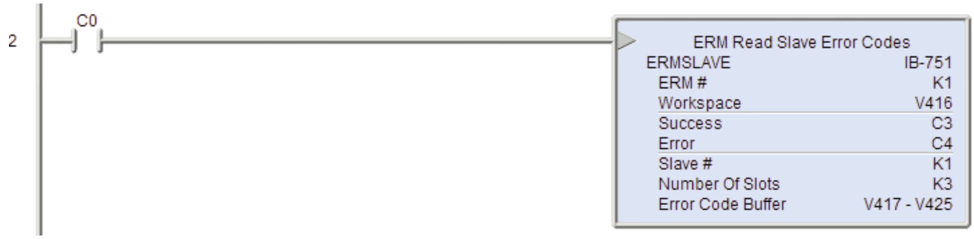
Rung 1: The ERM Config IBox is responsible for coordination/interlocking of all ERM type IBoxes for one specific ERM module. Tag the ERM in slot 1 as ERM# K1. All other ERxxxx IBoxes refer to this module # as K1. If you need to move the module in the base to a different slot, then you only need to change this one IBox.



ERM Config

ERM	**IB-750**
ERM #	K1
Slot	K1
Workspace	V400

Rung 2: The error information will be read from ERM #1 with the result placed into seven memory locations starting at V417. C3 will be enable if the read is a success, C4 will be enabled if the attempted read results in failure.

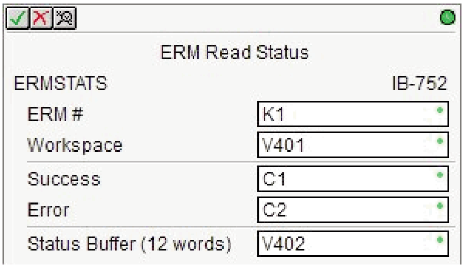


ERM Read Slave Error Codes

ERMSLAVE	IB-751
ERM #	K1
Workspace	V416
Success	C3
Error	C4
Slave #	K1
Number Of Slots	K3
Error Code Buffer	V417 - V425

ERM Read Status (ERMSTATS) (IB-752)

DS6 **ONLY**	Used
HPP	N/A



ERM Read Status

ERMSTATS	IB-752
ERM #	K1
Workspace	V401
Success	C1
Error	C2
Status Buffer (12 words)	V402

The ERM Read Status IBox will retrieve runtime status data from the ERM or ERM100.

When the PLC is in Run mode, the ERM or ERM100 module will compute some statistical data describing the ERM network's performance. These status values can be used to monitor the health of the backplane interface between the CPU and the ERM or ERM100 module, and to monitor the health of the Ethernet network connecting the ERM or ERM100 to its slaves.

This IBox references the ERM # of the ERM Config IBox that is controlling the ERM or ERM100 module in a specific slot. The ERM Config contains built-in interlocking logic that is used to synchronize the processing of this IBox with all of the other IBoxes in the ladder program that are being processed by the same ERM or ERM100 module.

The ERM Read Status IBox retrieves 7 status values from the ERM or ERM100 and places those values in consecutive V-Memory locations. The values of these status registers will reset to 0 on each Program mode -to- Run mode change. The definitions of those status values follows:

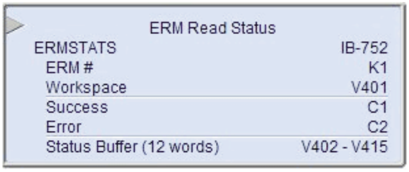
Number	Size	Format	Name	Description
1	Word	Decimal	Minimum I/O Scan	The minimum amount of time (in milliseconds) the ERM or ERM100 module spent updating all of its Ethernet slaves.
2	Word	Decimal	Maximum I/O Scan	The maximum amount of time (in milliseconds) the ERM or ERM100 module spent updating all of its Ethernet slaves.
3	DWord	Decimal	Total Time	The amount of time (in milliseconds) the ERM or ERM100 module has been running.
4	DWord	Decimal	Number of I/O Scans	The total number of I/O scans the ERM or ERM100 has completed.
5	DWord	Decimal	Number of PLC Read Retries	The total number of retries on Read Requests that the ERM or ERM100 module has generated when communicating across the backplane to the CPU.
6	DWord	Decimal	Number of PLC Write Retries	The total number of retries on Write Requests that the ERM or ERM100 module has generated when communicating across the backplane to the CPU.
7	DWord	Decimal	Number of Slave Retries	The total number of retries on Ethernet Read and Write Requests that the ERM or ERM100 module has generated when communicating with its slaves.

ERMSTATS Parameters

- ERM#: A reference number or resource number used to uniquely identify the ERM network.
- Workspace: A V-Memory register that is used internally by this IBox. It must not be used by any other instructions in the PLC.
- Success: This BIT will be ON if the Read Status succeeds and OFF if the Read Status fails.
- Error: This BIT will be OFF if the Read Status succeeds and ON if the Read Status fails.
- Status Buffer (12 words): The first of the 12 consecutive V-Memory registers where the ERM Status values will be stored.

Parameter	DL405 Range
ERM# . . . . K	K0-255
Workspace V	All User V Memory
Success . . . . X,Y,C,GX,GY,B	All Bit Memory
Error . . . . X,Y,C,GX,GY,B	All Bit Memory
Status Buffer V	All User V Memory

Note:*The gray triangle at the right end of an input leg indicates the input is edge triggered. Meaning that each time the input logic transitions from OFF to ON this instruction will execute.*



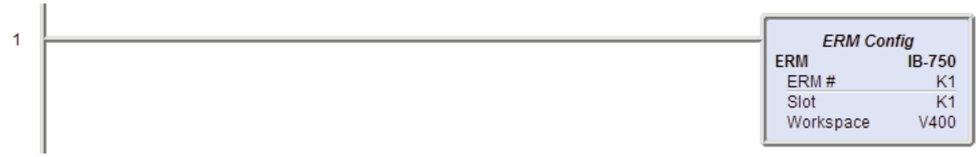
ERM Read Status

ERMSTATS	IB-752
ERM #	K1
Workspace	V401
Success	C1
Error	C2
Status Buffer (12 words)	V402 - V415

With each execution, this instruction will run to completion even if the input logic transitions to OFF before the instruction completes.

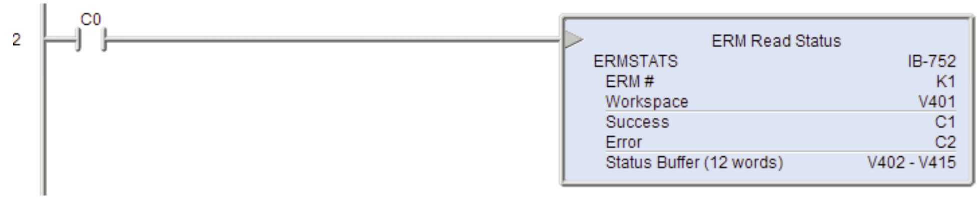
ERMSTATS Example

Rung 1: The ERM Config IBox is responsible for coordination/interlocking of all ERM type IBoxes for one specific ERM module. Tag the ERM in slot 1 as ERM# K1. All other ERxxxx IBoxes refer to this module # as K1. If you need to move the module in the base to a different slot, then you only need to change this one IBox.



ERM Config
ERM **IB-750**
ERM # K1
Slot K1
Workspace V400

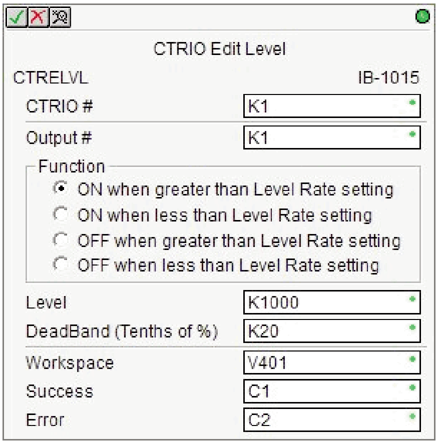
Rung 2: The error information will be read from ERM #1 with the result placed into twelve memory locations starting at V402. C1 will be enable if the read is a success, C2 will be enabled if the attempted read results in failure.



ERM Read Status
ERMSTATS IB-752
ERM # K1
Workspace V401
Success C1
Error C2
Status Buffer (12 words) V402 - V415

CTRIO Edit Level (CTRELVL) (IB-1015)

DS6 **ONLY**	Used
HPP	N/A



CTRIO Edit Level
CTRELVL IB-1015
CTRIO # K1
Output # K1
Function
● ON when greater than Level Rate setting
○ ON when less than Level Rate setting
○ OFF when greater than Level Rate setting
○ OFF when less than Level Rate setting
Level K1000
DeadBand (Tenths of %) K20
Workspace V401
Success C1
Error C2

The CTRIO Edit Level IBox will configure the Level Mode behavior for a Discrete Output of a CTRIO module.

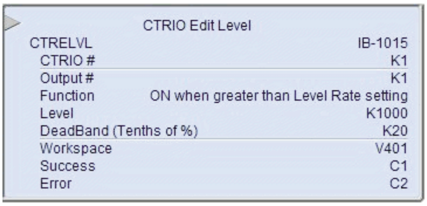
It references the CTRIO # in the CTRIO Config IBox that is controlling the CTRIO module.

CTRELVL Parameters

- CTRIO#: This number corresponds to the CTRIO # specified in the CTRIO Config IBox for the CTRIO module being used.
- Output #: Identifies which CTRIO Output to configure.
- Function (selectable option): ON when greater than Level Rate Setting/ON when less than Level Rate Setting/OFF when greater than Level Rate Setting/OFF when less than Level Rate Setting.
- Level: The DWORD count value at which the Function above will be active (decimal).
- Deadband (Tenths of %): The value above and below the Level at which the Function will be active (BCD).
- Workspace: A V-Memory register that is used internally by this IBox. It must not be used by any other instructions in the PLC.
- Success: This BIT will be ON if the Edit Level succeeds and OFF if the Edit Level fails.
- Error: This BIT will be OFF if the Edit Level succeeds and ON if the Edit Level fails.

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Level . . . . V,K	K0-2147483647, All User V Memory
Deadband# . . . . V,K	K0-1000, All User V Memory
Workspace V	All User V Memory
Success . . . . X,Y,C,GX,GY,B	All Bit Memory
Error . . . . X,Y,C,GX,GY,B	All Bit Memory

Note:*The gray triangle at the right end of an input leg indicates the input is edge triggered. Meaning that each time the input logic transitions from OFF to ON this instruction will execute.*

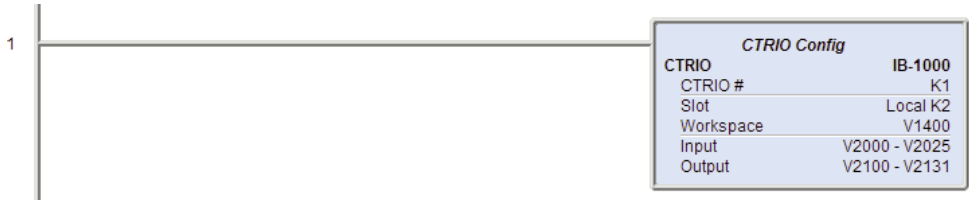


CTRIO Edit Level
CTRELVL IB-1015
CTRIO # K1
Output # K1
Function ON when greater than Level Rate setting
Level K1000
DeadBand (Tenths of %) K20
Workspace V401
Success C1
Error C2

With each execution, this instruction will run to completion even if the input logic transitions to OFF before the instruction completes.

CTRELVL Example

Rung 1: This sets up the CTRIO module in slot 2 of the base. Each CTRIO module in the system will need a separate CTRIO Config IBox before any CTRxxxx IBoxes can be used. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2100 through V2131 for its output data.

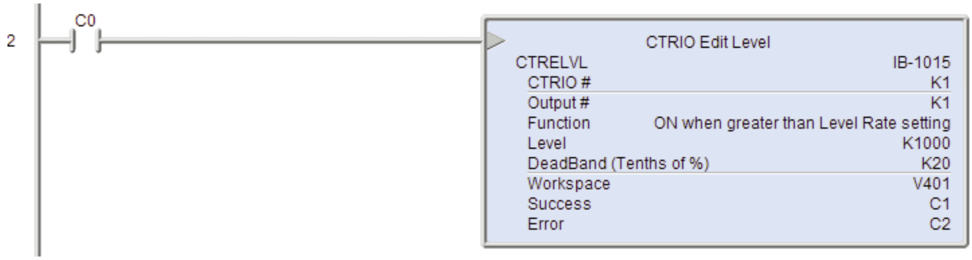


1

CTRIO Config

CTRIO	IB-1000
CTRIO #	K1
Slot	Local K2
Workspace	V1400
Input	V2000 - V2025
Output	V2100 - V2131

Rung 2: This rung is a sample method for configuring the level behavior of a CTRIO output. Turning on C0 will cause the CTRELVL instruction to set the first output of the module to ON when the level setting of K1000 is exceeded. If the level request is successful, C1 will turn ON. If the level request fails, C2 will turn ON.



2 C0

CTRIO Edit Level

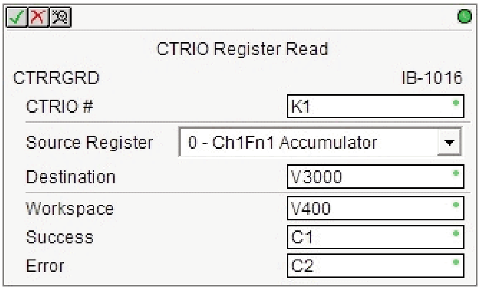
CTRELVL	IB-1015
CTRIO #	K1
Output #	K1
Function	ON when greater than Level Rate setting
Level	K1000
DeadBand (Tenths of %)	K20
Workspace	V401
Success	C1
Error	C2

CTRIO Register Read (CTRRGRD) (IB-1016)

DS6 **ONLY**	Used
HPP	N/A

The CTRIO Register Read IBox will retrieve the value from the specified register in a CTRIO or CTRIO2 module.

It references the CTRIO # in the CTRIO Config IBox that is controlling the CTRIO module.



CTRIO Register Read

CTRRGRD IB-1016

CTRIO # K1

Source Register 0 - Ch1Fn1 Accumulator

Destination V3000

Workspace V400

Success C1

Error C2

CTRRGRD Parameters

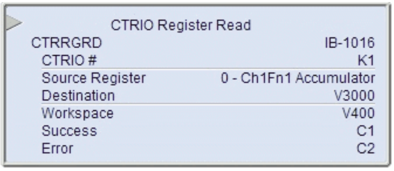
- CTRIO#: This number corresponds to the CTRIO # specified in the CTRIO Config IBox for the CTRIO module being used.
- Source Register (selectable option):

0 - Ch1Fn1 Accumulator	10 - Ch2Fn1 Reset Value
1 - Ch1Fn2 Accumulator	11 - Ch2Fn2 Reset Value
2 - Ch2Fn1 Accumulator	12 - Ch1A Filter Time (CTRIO2)
3 - Ch2Fn2 Accumulator	13 - Ch1B Filter Time (CTRIO2)
4 - Out0 Position	14 - Ch1C Filter Time (CTRIO2)
5 - Out1 Position	15 - Ch1D Filter Time (CTRIO2)
6 - Out2 Position	16 - Ch2A Filter Time (CTRIO2)
7 - Out3 Position	17 - Ch2B Filter Time (CTRIO2)
8 - Ch1Fn1 Reset Value	18 - Ch2C Filter Time (CTRIO2)
9 - Ch1Fn2 Reset Value	19 - Ch2D Filter Time (CTRIO2)

- Destination: A DWORD that is used to store the value read from the specified register.
- Workspace: A V-Memory register that is used internally by this IBox. It must not be used by any other instructions in the PLC.
- Success: This BIT will be ON if the Register Read succeeds and OFF if the Register Read fails.
- Error: This BIT will be OFF if the Register Read succeeds and ON if the Register Read fails.

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Destination V	All User V Memory
Workspace V	All User V Memory
Success . . . . X,Y,C,GX,GY,B	All Bit Memory
Error . . . . X,Y,C,GX,GY,B	All Bit Memory

***Note:** The gray triangle at the right end of an input leg indicates the input is edge triggered. Meaning that each time the input logic transitions from OFF to ON this instruction will execute.*

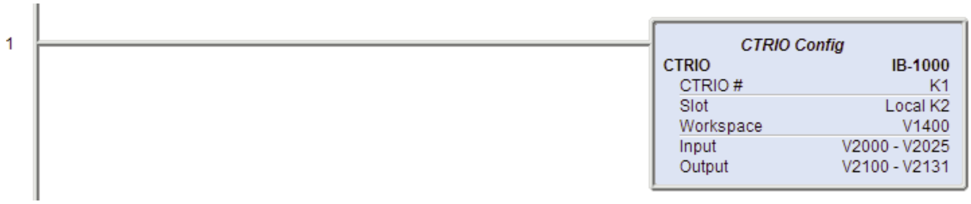


CTRIO Register Read	
CTRRGRD	IB-1016
CTRIO #	K1
Source Register	0 - Ch1Fn1 Accumulator
Destination	V3000
Workspace	V400
Success	C1
Error	C2

With each execution, this instruction will run to completion even if the input logic transitions to OFF before the instruction completes.

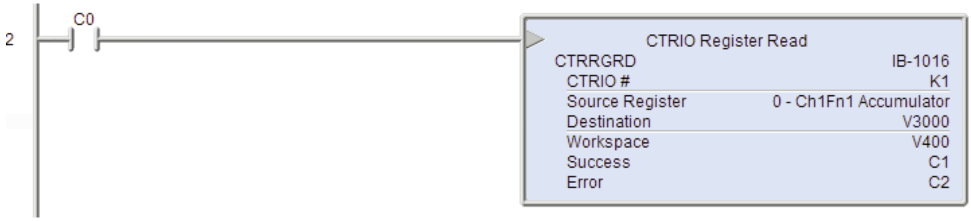
CTRRGRD Example

Rung 1: This sets up the CTRIO module in slot 2 of the base. Each CTRIO module in the system will need a separate CTRIO Config IBox before any CTRxxxx IBoxes can be used. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2100 through V2131 for its output data.



CTRIO Config	
CTRIO	IB-1000
CTRIO #	K1
Slot	Local K2
Workspace	V1400
Input	V2000 - V2025
Output	V2100 - V2131

Rung 2: This rung is a sample method for reading a register of a CTRIO module. Turning on C0 will cause the CTRRGRD instruction to read the Channel 1 Function 1 register and store the result in V3000-V3001. If the register read request is successful, C1 will turn ON. If the register read request fails, C2 will turn ON.



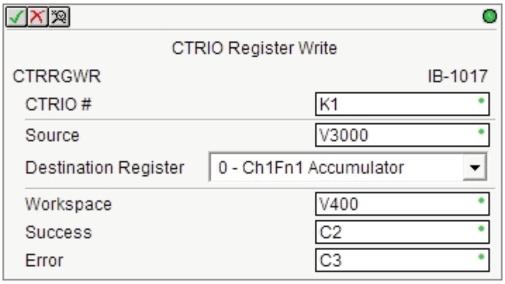
CTRIO Register Read	
CTRRGRD	IB-1016
CTRIO #	K1
Source Register	0 - Ch1Fn1 Accumulator
Destination	V3000
Workspace	V400
Success	C1
Error	C2

CTRIO Register Write (CTRRGWR) (IB-1017)

DS6 **ONLY**	Used
HPP	N/A

The CTRIO Register Write IBox will write the specified value to the selected register in a CTRIO or CTRIO2 module.

It references the CTRIO # in the CTRIO Config IBox that is controlling the CTRIO module.



CTRRGWR IB-1017

CTRIO #: K1

Source: V3000

Destination Register: 0 - Ch1Fn1 Accumulator

Workspace: V400

Success: C2

Error: C3

CTRRGWR Parameters

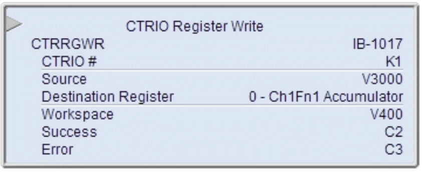
- CTRIO#: This number corresponds to the CTRIO # specified in the CTRIO Config IBox for the CTRIO module being used.
- Source: A DWORD that contains the value or a Hex constant value to write to the specified register.
- Destination Register (selectable option):

0 - Ch1Fn1 Accumulator	10 - Ch2Fn1 Reset Value
1 - Ch1Fn2 Accumulator	11 - Ch2Fn2 Reset Value
2 - Ch2Fn1 Accumulator	12 - Ch1A Filter Time (CTRIO2)
3 - Ch2Fn2 Accumulator	13 - Ch1B Filter Time (CTRIO2)
4 - Out0 Position	14 - Ch1C Filter Time (CTRIO2)
5 - Out1 Position	15 - Ch1D Filter Time (CTRIO2)
6 - Out2 Position	16 - Ch2A Filter Time (CTRIO2)
7 - Out3 Position	17 - Ch2B Filter Time (CTRIO2)
8 - Ch1Fn1 Reset Value	18 - Ch2C Filter Time (CTRIO2)
9 - Ch1Fn2 Reset Value	19 - Ch2D Filter Time (CTRIO2)

- Workspace: A V-Memory register that is used internally by this IBox. It must not be used by any other instructions in the PLC.
- Success: This BIT will be ON if the Register Write succeeds and OFF if the Register Write fails.
- Error: This BIT will be OFF if the Register Write succeeds and ON if the Register Write fails.

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Source . . . . K,V	K0-FFFFFFFF, All V Memory
Workspace V	All User V Memory
Success . . . . X,Y,C,GX,GY,B	All Bit Memory
Error . . . . X,Y,C,GX,GY,B	All Bit Memory

***Note:**The gray triangle at the right end of an input leg indicates the input is edge triggered. Meaning that each time the input logic transitions from OFF to ON this instruction will execute.*

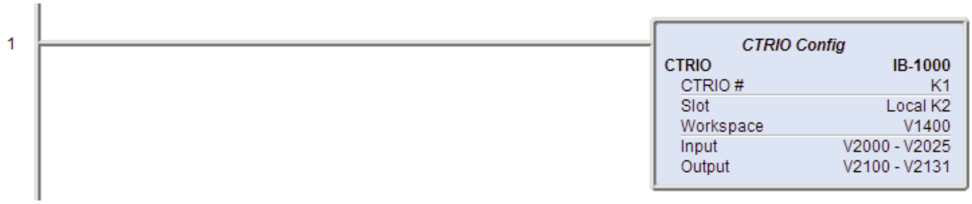


CTRIO Register Write
CTRRGWR IB-1017
CTRIO # K1
Source V3000
Destination Register 0 - Ch1Fn1 Accumulator
Workspace V400
Success C2
Error C3

With each execution, this instruction will run to completion even if the input logic transitions to OFF before the instruction completes.

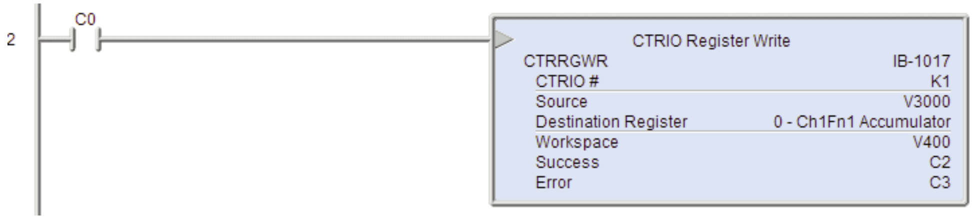
CTRRGWR Example

Rung 1: This sets up the CTRIO module in slot 2 of the base. Each CTRIO module in the system will need a separate CTRIO Config IBox before any CTRxxxx IBoxes can be used. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2100 through V2131 for its output data.



CTRIO Config
CTRIO IB-1000
CTRIO # K1
Slot Local K2
Workspace V1400
Input V2000 - V2025
Output V2100 - V2131

Rung 2: This rung is a sample method for writing a register of a CTRIO module. Turning on C0 will cause the CTRRGWR instruction to write the value stored in V3000-V3001 to the Channel 1 Function 1 accumulator register. If the register write request is successful, C2 will turn ON. If the register write request fails, C3 will turn ON.



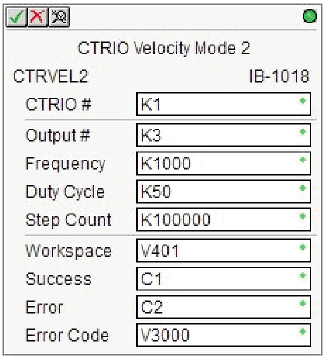
CTRIO Register Write
CTRRGWR IB-1017
CTRIO # K1
Source V3000
Destination Register 0 - Ch1Fn1 Accumulator
Workspace V400
Success C2
Error C3

CTRIO Velocity Mode 2 (CTRVEL2) (IB-1018)

DS6 **ONLY**	Used
HPP	N/A

The CTRIO Velocity Mode 2 IBox will setup the CTRIO or CTRIO2 module to perform a Velocity Mode operation on the specified CTRIO output. This runtime function generates the desired number of output pulses as defined by the frequency and duty cycle. A Step Count value of -1 instructs the CTRIO to continuously generate output pulses.

The specified CTRIO output must already be configured as a Pulse Output. This configuration is done via CTRIO Workbench.



CTRIO Velocity Mode 2

CTRVEL2 IB-1018

CTRIO # K1

Output # K3

Frequency K1000

Duty Cycle K50

Step Count K100000

Workspace V401

Success C1

Error C2

Error Code V3000

The CTRIO Velocity Mode IBox will take multiple PLC scans to complete. Each time this IBox is triggered it will run to completion exactly one time. It will start running on the rising edge of the input circuit and once triggered, it will run to completion. Any rising edges generated before the IBox completes will be ignored. The IBox is complete when the either the Success bit or Error bit are set ON.

It references the CTRIO # in the CTRIO Config IBox that is controlling the CTRIO module.

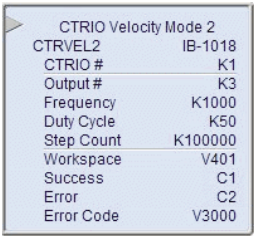
CTRVEL2 Parameters

- CTRIO#: This number corresponds to the CTRIO # specified in the CTRIO Config IBox for the CTRIO module being used.
- Output#: Identifies which CTRIO Output to configure.
- Frequency: Specifies the pulse output frequency in Hertz.
- Duty Cycle: Specifies the duty cycle of the output pulses (0 = 50%).
- Step Count: This DWORD value specifies the number of pulses to output. A Step Count value of -1 (or 0xFFFFFFFF) causes the CTRIO to output pulses continuously. Negative Step Count values must be V-Memory references.
- Workspace: A V-Memory register that is used internally by this IBox. It must not be used by any other instructions in the PLC.
- Success: This BIT will be ON if the Setup Velocity Mode succeeds and OFF if it fails.
- Error: This BIT will be OFF if the Setup Velocity Mode succeeds and ON if it fails.
- Error Code: A V-Memory register that is used to store the Error if the Setup Velocity Mode fails. The following table has a list of the possible Error Code values:

Error Code	Description
0	No Error
2002	Output Enable was already ON when the Instruction was enabled.
2003	The CTRIO module reported an error. Use the CTRIO Read Error (CTRRDER) IBox to read the CTRIO module's error code to determine what went wrong.

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Frequency . . . . V,K	K20-20000, K20-65535 (CTRIO2), All User V Memory
Duty Cycle . . . . V,K	K0-99, All User V Memory
Step Count . . . . K,V	K0-2147483647, All User V Memory
Workspace V	All User V Memory
Success . . . . X,Y,C,GX,GY,B	All Bit Memory
Error . . . . X,Y,C,GX,GY,B	All Bit Memory
Error Code V	All V Memory

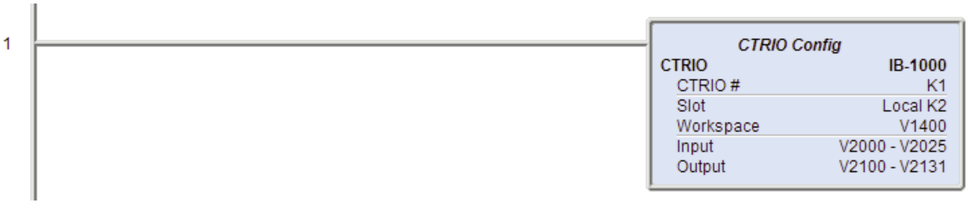
Note: *The gray triangle at the right end of an input leg indicates the input is edge triggered. Meaning that each time the input logic transitions from OFF to ON this instruction will execute.*


CTRIO Velocity Mode 2
CTRVEL2 IB-1018
CTRIO # K1
Output # K3
Frequency K1000
Duty Cycle K50
Step Count K100000
Workspace V401
Success C1
Error C2
Error Code V3000

With each execution, this instruction will run to completion even if the input logic transitions to OFF before the instruction completes.

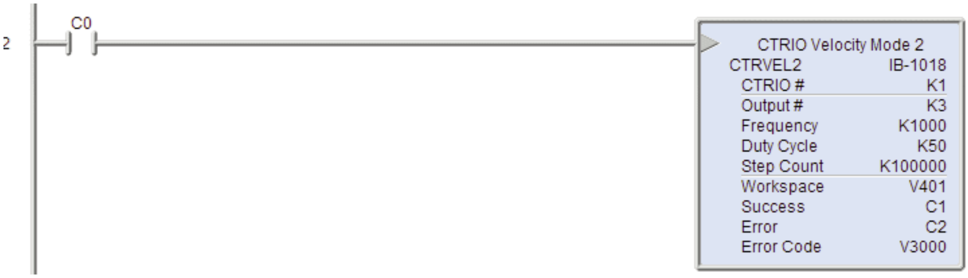
CTRVEL2 Example

Rung 1: This sets up the CTRIO module in slot 2 of the base. Each CTRIO module in the system will need a separate CTRIO Config IBox before any CTRxxxx IBoxes can be used. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2100 through V2131 for its output data.



CTRIO Config
CTRIO IB-1000
CTRIO # K1
Slot Local K2
Workspace V1400
Input V2000 - V2025
Output V2100 - V2131

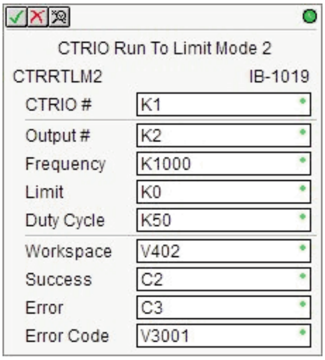
Rung 2: This CTRIO Velocity Mode 2 IBox sets up Output #3 in CTRIO #1 to output 100,000 pulses at a Frequency of 1000 Hz with a 50% Duty Cycle.



CTRIO Velocity Mode 2
CTRVEL2 IB-1018
CTRIO # K1
Output # K3
Frequency K1000
Duty Cycle K50
Step Count K100000
Workspace V401
Success C1
Error C2
Error Code V3000

CTRIO Run to Limit Mode 2 (CTRRTLM2) (IB-1019)

DS6 **ONLY**	Used
HPP	N/A



CTRIO Run To Limit Mode 2

CTRRTLM2 IB-1019

CTRIO #	K1
Output #	K2
Frequency	K1000
Limit	K0
Duty Cycle	K50
Workspace	V402
Success	C2
Error	C3
Error Code	V3001

The CTRIO Run to Limit Mode 2 IBox will setup the CTRIO or CTRIO2 module to perform a Run to Limit Mode operation on the specified CTRIO output.

The specified CTRIO Output must already be configured as a Pulse Output and the specified Input must already be configured as a Limit. This configuration is done via CTRIO Workbench.

The CTRIO Run To Limit Mode IBox will take multiple PLC scans to complete. Each time this IBox is triggered it will run to completion exactly one time. It will start running on the rising edge of the input circuit and once triggered, it will run to completion. Any rising edges generated before the IBox completes will be ignored. The IBox is complete when the either the Success bit or Error bit are set ON.

It references the CTRIO # in the CTRIO Config IBox that is controlling the CTRIO module.

CTRRTLM2 Parameters

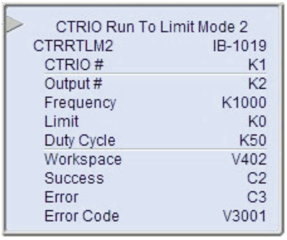
- CTRIO#: This number corresponds to the CTRIO # specified in the CTRIO Config IBox for the CTRIO module being used.
- Output#: Identifies which CTRIO Output to configure.
- Frequency: Specifies the pulse output frequency in Hertz.
- Limit: Specifies which CTRIO Input resource is the Limit and which level of that Limit to use. See the table on right for a list of the valid Limit values.
- Duty Cycle: Specifies the duty cycle of the output pulses (0 = 50%).
- Workspace: A V-Memory register that is used internally by this IBox. It must not be used by any other instructions in the PLC.
- Success: This BIT will be ON if the Run to Limit succeeds and OFF if it fails.
- Error: This BIT will be OFF if the Run to Limit succeeds and ON if it fails.
- Error Code: A V-Memory register that is used to store the Error if the Run to Limit fails. The following table has a list of the possible Error Code values.

Value	Description
00	Ch1/C High (ON)
10	Ch1/C Low (OFF)
01	Ch1/D High (ON)
11	Ch1/D Low (OFF)
02	Ch2/C High (ON)
12	Ch2/C Low (OFF)
03	Ch2/D High (ON)
13	Ch2/D Low (OFF)

Error Code	Description
0	No Error
2002	Output Enable was already ON when the Instruction was enabled.
2003	The CTRIO module reported an error. Use the CTRIO Read Error (CTRRDER) IBox to read the CTRIO module's error code to determine what went wrong.

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Frequency . . . . V,K	K20-20000, K20-65535 (CTRIO2), All User V Memory
Limit . . . . V,K	K0-FF, All User V Memory
Duty Cycle . . . . V,K	K0-99, All User V Memory
Workspace V	All User V Memory
Success . . . . X,Y,C,GX,GY,B	All Bit Memory
Error . . . . X,Y,C,GX,GY,B	All Bit Memory
Error Code V	All V Memory

Note: *The gray triangle at the right end of an input leg indicates the input is edge triggered. Meaning that each time the input logic transitions from OFF to ON this instruction will execute.*



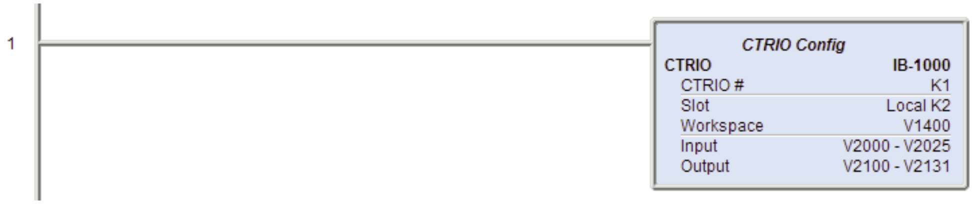
CTRIO Run To Limit Mode 2

CTRRTLM2	IB-1019
CTRIO #	K1
Output #	K2
Frequency	K1000
Limit	K0
Duty Cycle	K50
Workspace	V402
Success	C2
Error	C3
Error Code	V3001

With each execution, this instruction will run to completion even if the input logic transitions to OFF before the instruction completes.

CTRRTLM2 Example

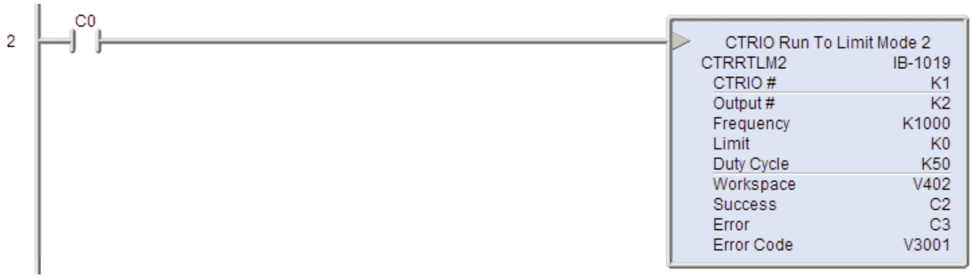
Rung 1: This sets up the CTRIO module in slot 2 of the base. Each CTRIO module in the system will need a separate CTRIO Config IBox before any CTRxxxx IBoxes can be used. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2100 through V2131 for its output data.



CTRIO Config

CTRIO	IB-1000
CTRIO #	K1
Slot	Local K2
Workspace	V1400
Input	V2000 - V2025
Output	V2100 - V2131

Rung 2: This CTRIO Run To Limit Mode 2 IBox sets up Output #2 in CTRIO #1 to output pulses at a Frequency of 1000 Hz with a 50% Duty Cycle until Limit #0 comes ON.

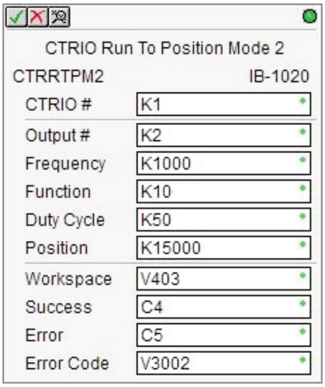


CTRIO Run To Limit Mode 2

CTRRTLM2	IB-1019
CTRIO #	K1
Output #	K2
Frequency	K1000
Limit	K0
Duty Cycle	K50
Workspace	V402
Success	C2
Error	C3
Error Code	V3001

CTRIO Run to Position Mode 2 (CTRRTPM2) (IB-1020)

DS6 **ONLY**	Used
HPP	N/A



CTRIO Run To Position Mode 2

CTRRTPM2 IB-1020

CTRIO #	K1
Output #	K2
Frequency	K1000
Function	K10
Duty Cycle	K50
Position	K15000
Workspace	V403
Success	C4
Error	C5
Error Code	V3002

The CTRIO Run to Position Mode 2 IBox will setup the CTRIO or CTRIO2 module to perform a Run to Position Mode operation on the specified CTRIO output.

The specified CTRIO Output must already be configured as a Pulse Output and the specified Input must already be configured as a Counter or Quad Counter. This configuration is done via CTRIO Workbench.

The CTRIO Run To Position Mode IBox will take multiple PLC scans to complete. Each time this IBox is triggered it will run to completion exactly one time. It will start running on the rising edge of the input circuit and once triggered, it will run to completion. Any rising edges generated before the IBox completes will be ignored. The IBox is complete when the either the Success bit or Error bit are set ON.

It references the CTRIO # in the CTRIO Config IBox that is controlling the CTRIO module.

CTRRTPM2 Parameters

- CTRIO#: This number corresponds to the CTRIO # specified in the CTRIO Config IBox for the CTRIO module being used.
- Output#: Identifies which CTRIO Output to configure.
- Frequency: Specifies the pulse output frequency in Hertz.
- Function: Specifies which CTRIO Input resource and the comparison operator that determines when the target position is reached. The following is a list of the valid resource/comparison operators:

Value	Description
00	Less Than Ch1/Fn1
10	Greater Than Ch1/Fn1
01	Less Than Ch1/Fn2
11	Greater Than Ch1/Fn2
02	Less Than Ch2/Fn1
12	Greater Than Ch2/Fn1
03	Less Than Ch2/Fn2
13	Greater Than Ch2/Fn2

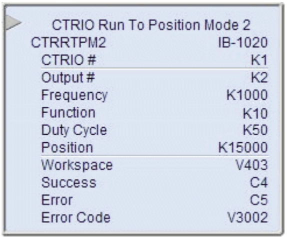
- Duty Cycle: Specifies the duty cycle of the output pulses (0 = 50%).
- Position: This DWORD value specifies the target position. Positive/Negative target position values are used in concert with the Greater-than/Less-than comparison operators to determine when the target position has been reached. Negative target position values must be V-Memory references.

- Workspace: A V-Memory register that is used internally by this IBox. It must not be used by any other instructions in the PLC.
- Success: This BIT will be ON if the Setup Run to Position succeeds and OFF if it fails.
- Error: This BIT will be OFF if the Setup Run To Position succeeds and ON if it fails.
- Error Code: A V-Memory register that is used to store the Error if the Run to Position fails. The following table has a list of the possible Error Code values:

Error Code	Description
0	No Error
2002	Output Enable was already ON when the Instruction was enabled.
2003	The CTRIO module reported an error. Use the CTRIO Read Error (CTRRDER) IBox to read the CTRIO module's error code to determine what went wrong.

Parameter	DL405 Range
CTRIO# . . . . K	K0-255
Output# . . . . K	K0-3
Frequency . . . . V,K	K20-20000, K20-65535 (CTRIO2), All User V Memory
Function . . . . V,K	See table on previous page, All User V Memory
Duty Cycle . . . . V,K	K0-99, All User V Memory
Position . . . . V,K	K0-2147434528, All User V Memory
Workspace V	All User V Memory
Success . . . . X,Y,C,GX,GY,B	All Bit Memory
Error . . . . X,Y,C,GX,GY,B	All Bit Memory
Error Code V	All V Memory

Note: *The gray triangle at the right end of an input leg indicates the input is edge triggered. Meaning that each time the input logic transitions from OFF to ON this instruction will execute.*



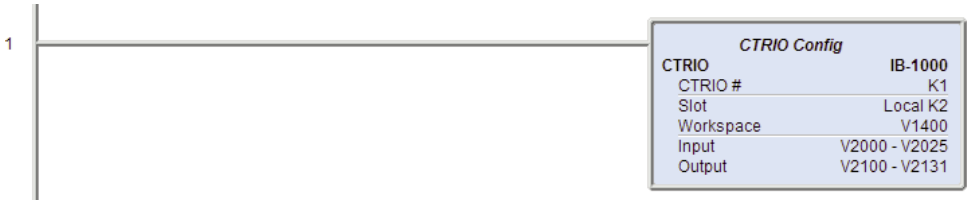
CTRIO Run To Position Mode 2

CTRRTPM2	IB-1020
CTRIO #	K1
Output #	K2
Frequency	K1000
Function	K10
Duty Cycle	K50
Position	K15000
Workspace	V403
Success	C4
Error	C5
Error Code	V3002

With each execution, this instruction will run to completion even if the input logic transitions to OFF before the instruction completes.

CTRRTPM2 Example

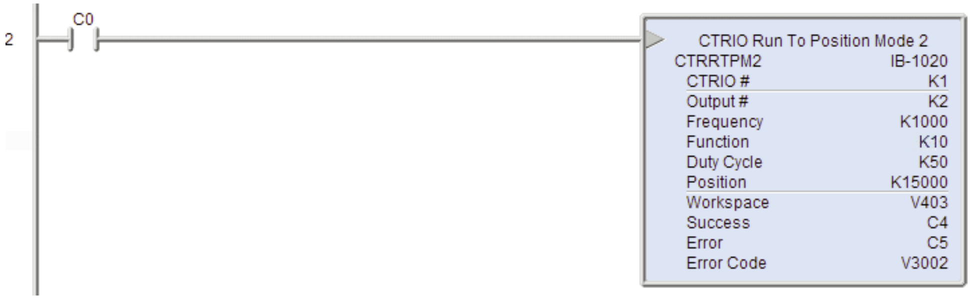
Rung 1: This sets up the CTRIO module in slot 2 of the base. Each CTRIO module in the system will need a separate CTRIO Config IBox before any CTRxxxx IBoxes can be used. The CTRIO has been configured to use V2000 through V2025 for its input data, and V2100 through V2131 for its output data.



CTRIO Config

CTRIO	**IB-1000**
CTRIO #	K1
Slot	Local K2
Workspace	V1400
Input	V2000 - V2025
Output	V2100 - V2131

Rung 2: This CTRIO Run To Position Mode 2 IBox sets up Output #2 in CTRIO #1 to output pulses at a Frequency of 1000 Hz with a 50% Duty Cycle, use the 'Greater than Ch1/Fn1' comparison operator, until the input position of 15,000 is reached.



CTRIO Run To Position Mode 2

CTRRTPM2	IB-1020
CTRIO #	K1
Output #	K2
Frequency	K1000
Function	K10
Duty Cycle	K50
Position	K15000
Workspace	V403
Success	C4
Error	C5
Error Code	V3002