

DL205 Instruction Set

Math Instructions (Accumulator)

- Add (ADD)**
Adds a BCD value in the lower 16 bits in the accumulator with a V-memory location. The result resides in the accumulator.
- Add Double (ADDD)**
Adds a BCD value in the accumulator with two consecutive V-memory locations or an 8-digit constant. The result resides in the accumulator.
- Add Real Number (ADDR)**
D2-250-1, D2-260 only. Adds a real number in the accumulator with a real number constant or a real number contained in two consecutive V-memory locations. The result resides in the accumulator.
- Subtract (SUB)**
Subtracts a BCD value, which is either a V-memory location or a 4-digit constant from the lower 16 bits in the accumulator. The result resides in the accumulator.
- Subtract Double (SUBD)**
Subtracts a BCD value, which is either two consecutive V-memory locations or an 8-bit constant, from a value in the accumulator. The result resides in the accumulator.
- Subtract Real Number (SUBR)**
D2-250-1, D2-260 only. Subtracts a real number, which is either two consecutive V-memory locations or an 8-digit constant, from the real number in the accumulator. The result resides in the accumulator.
- Multiply (MUL)**
Multiplies a BCD value, which is either a V-memory location or a 4-digit constant, by the value in the lower 16 bits in the accumulator. The result resides in the accumulator.
- Multiply Double (MULD)**
D2-250-1, D2-260 only. Multiplies a BCD value contained in two consecutive V-memory locations by the value in the accumulator. The result resides in the accumulator.
- Multiply Real Number (MULR)**
D2-250-1, D2-260 only. Multiplies a real number, which is either two consecutive V-memory locations or a real number constant, by the real number in the accumulator. The result resides in the accumulator.
- Divide (DIV)**
Divides a BCD value in the accumulator by a BCD value which is either a V-memory location or a 4-digit constant. The result resides in the accumulator.
- Divide Double (DIVD)**
D2-250-1, D2-260 only. Divides a BCD value in the accumulator by a BCD value which is either two consecutive V-memory locations or a 4-digit constant. The result resides in the accumulator.
- Divides Real Number (DIVR)**
D2-250-1, D2-260 only. Divides a real number in the accumulator by a real number which is either two consecutive V-memory locations or a real number constant. The result resides in the accumulator.
- Increment (INC)**
D2-250-1, D2-260 only. Increments a BCD value in a specified V-memory location by 1 each time the instruction is executed.
- Decrement (DEC)**
D2-250-1, D2-260 only. Decrements a BCD value in a specified V-memory location by 1 each time the instruction is executed.
- Add Binary (ADDB)**
D2-250-1, D2-260 only. Adds the binary value in the lower 16 bits of the accumulator to a value which is either a V-memory location or a 16-bit constant. The result resides in the accumulator.
- Add Binary Double (ADDBD)**
D2-260 only. Adds the binary value in the accumulator to a value which is either two consecutive V-memory locations or a 32-bit constant. The result resides in the accumulator.
- Subtract Binary (SUBB)**
D2-250-1, D2-260 only. Subtracts a 16-bit binary value, which is either a V-memory location or a 16-bit constant, from the lower 16 bits in the accumulator. The result resides in the accumulator.
- Subtract Binary Double (SUBBD)**
D2-260 only. Subtracts a 32-bit binary value, which is either two consecutive V-memory locations or a 32-bit constant, from the value in the accumulator. The result resides in the accumulator.
- Multiply Binary (MULB)**
D2-250-1, D2-260 only. Multiplies a 16-bit binary value, which is either a V-memory location or a 16-bit constant, by the lower 16 bits in the accumulator. The result resides in the accumulator.
- Divide Binary (DIVB)**
D2-250-1, D2-260 only. Divides the binary value in the lower 16 bits in the accumulator by a value which is either a V-memory location or a 16-bit constant. The result resides in the accumulator.
- Increment Binary (INCB)**
Increments a binary value in a specified V-memory location by 1 each time the instruction is executed.
- Decrement Binary (DECB)**
Decrements a binary value in a specified V-memory location by 1 each time the instruction is executed.
- Add Formatted (ADDF)**
D2-260 only. Adds the BCD value in the accumulator to a value which is a range of discrete bits (1-32). The result resides in the accumulator.
- Subtract Formatted (SUBF)**
D2-260 only. Subtracts a BCD value which is a range of discrete bits (1-32) from the BCD value in the accumulator. The result resides in the accumulator.
- Multiply Formatted (MULF)**
D2-260 only. Multiplies a BCD value in the lower 16-bits in the accumulator by a BCD value which is a range of discrete bits (1-16). The result resides in the accumulator.
- Divide Formatted (DIVF)**
D2-260 only. Divides the BCD value in the lower 16-bits in the accumulator by the BCD value which is a range of discrete bits (1-16). The result resides in the accumulator.
- Add Top of Stack (ADDS)**
D2-260 only. Adds the BCD value in the accumulator with the BCD value in the first level of the accumulator stack. The result resides in the accumulator.
- Subtract Top of Stack (SUBS)**
D2-260 only. Subtracts the BCD value in the first level of the accumulator stack from the BCD value in the accumulator. The result resides in the accumulator.
- Multiply Top of Stack (MULS)**
D2-260 only. Multiplies a 4-digit BCD value in the first level of the accumulator stack by a 4-digit BCD value in the accumulator. The result resides in the accumulator.

Divide by Top of Stack (DIVS)

- D2-260 only. Divides the 8-digit BCD value in the accumulator by the 4-digit BCD value in the first level of the accumulator by the 4-digit BCD value in the first level of the accumulator stack. The result resides in the accumulator.
- Add Binary Top of Stack (ADDBS)**
D2-260 only. Adds the binary value in the accumulator with the binary value in the first accumulator stack location. The result resides in the accumulator.
- Subtract Binary Top of Stack (SUBBS)**
D2-260 only. Subtracts the binary value in the first level of the accumulator stack from the binary value in the accumulator. The result resides in the accumulator.
- Multiply Binary Top of Stack (MULBS)**
D2-260 only. Multiplies the 16-bit binary value in the first level of the accumulator stack by the 16-bit binary value in the accumulator. The result resides in the accumulator.
- Divide Binary Top of Stack (DIVBS)**
D2-260 only. Divides a value in the accumulator by the binary value in the top location of the stack. The accumulator contains the result.

Transcendental Instructions (D2-260 only)

- Square Root Real (SQRTR)**
Takes the square root of the real number stored in the accumulator. The result resides in the accumulator.
- Sine Real (SINR)**
Takes the sine of the real number stored in the accumulator. The result resides in the accumulator.
- Cosine Real (COSR)**
Takes the cosine of the real number stored in the accumulator. The result resides in the accumulator.
- Tangent Real (TANR)**
Takes the tangent of the real number stored in the accumulator. The result resides in the accumulator.
- ARC Sine Real (ASINR)**
Takes the inverse sine of the real number stored in the accumulator. The result resides in the accumulator.
- ARC Cosine Real (ACOSR)**
Takes the inverse cosine of the real number stored in the accumulator. The result resides in the accumulator.
- ARC Tangent Real (ATANR)**
Takes the inverse tangent of the real number stored in the accumulator. The result resides in the accumulator.

Bit Instructions (Accumulator)

- Sum (SUM)**
D2-250-1, D2-260 only. Counts the number of bits set to "1" in the accumulator. The HEX result resides in the accumulator.
- Shift Left (SHFL)**
Shifts the bits in the accumulator a specified number of places to the left.
- Shift Right (SHFR)**
Shifts the bits in the accumulator a specified number of places to the right.
- Rotate Left (ROTL)**
D2-250-1, D2-260 only. Rotates the bits in the accumulator a specified number of places to the left.
- Rotate Right (ROTR)**
D2-250-1, D2-260 only. Rotates the bits in the accumulator a specified number of places to the right.
- Encode (ENCO)**
Encodes the bit position set to 1 in the accumulator, and returns the appropriate binary representation in the accumulator.
- Decodes (DECO)**
Decodes a 5 bit binary value (0-31) in the accumulator by setting the appropriate bit position to a 1. If the accumulator contains the value F (HEX), bit 15 will be set in the accumulator. If the value to be decoded is greater than 31, the number is divided by 32 until the value is less than 32 and then the value is decoded.

Number Conversion Instructions (Accumulator)

- Binary (BIN)**
Converts the BCD value in the accumulator to the equivalent binary value. The result resides in the accumulator.
- Binary Coded Decimal (BCD)**
Converts the binary value in the accumulator to the equivalent BCD value. The result resides in the accumulator.
- Invert (INV)**
Takes the one's complement of the 32-bit value in the accumulator. The result resides in the accumulator.
- Ten's Complement (BCDCPL)**
Takes the 10's complement (BCD) of the 8-digit accumulator.
- ASCII to HEX (ATH)**
D2-250-1, D2-260 only. Converts a table of ASCII values to a table of hexadecimal values.
- HEX to ASCII (HTA)**
D2-250-1, D2-260 only. Converts a table of hexadecimal values to a table of ASCII values.
- Segment (SEG)**
D2-250-1, D2-260 only. Converts four digit HEX value in accumulator to seven segment display format.
- Gray Code to BCD (GRAY)**
D2-240, D2-250-1, D2-260 only. Converts a 16-bit GRAY code value in the accumulator to a corresponding BCD value. The result resides in the accumulator.
- Shuffle Digits (SFLDGT)**
D2-240, D2-250-1, D2-260 only. Shuffles a maximum of 8 digits, rearranging them in a specified order. The result resides in the accumulator.
- Radian Real Conversion (RADR)**
D2-260 only. Converts the real degree value in the accumulator to the equivalent real number in radians. The result resides in the accumulator.
- Degree Real Conversion (DEGR)**
D2-260 only. Converts the real radian value in the accumulator to the equivalent real member of degrees. The result resides in the accumulator.
- Binary to Real Number (BTOR)**
D2-250-1, D2-260 only. Converts the binary value in the accumulator into a real number. The result resides in the accumulator.

Real to Binary (RTOB)

- D2-250-1, D2-260 only. Converts the real number in the accumulator into a binary value. The result resides in the accumulator.

Table Instructions

- Move (MOV)**
Moves the values from one V-memory table to another V-memory table.
- Move Memory Cartridge/Load Label (MOVMC/LDLBL)**
Copies data between V-memory and program ladder memory.
- Set Bit (SETBIT)**
D2-260 only. Sets a single bit (to a 0) in a V-memory location.
- Reset Bit (RSTBIT)**
D2-260 only. Resets a single bit (to a 0) in a V-memory location.

Extended Table Instructions (D2-260 only)

- Fill (FILL)**
Fills a table of specified V-memory locations with a value which is either a V-memory location or a 4-digit constant.
- Find (FIND)**
Finds a value in a V-memory table and returns the table position containing the value to the accumulator.
- Find Greater Than (FDGT)**
Finds a value in a V-memory table which is greater than the specified search value. The table position containing the value is returned to the accumulator.
- Find Block (FINDB)**
Finds a block of data values in a V-memory table and returns the starting address of the table containing the values to the accumulator.
- Table to Destination (TTD)**
Moves the value from the top of a V-memory table to a specified V-memory location. The table pointer increments each scan.
- Remove from Bottom (RFB)**
Moves the value from the bottom of a V-memory table to a specified V-memory location. The table pointer increments each scan.
- Source To Table (STT)**
Moves a value from a specified V-memory location to a V-memory table. The table pointer increments each scan.
- Remove from Top (RFT)**
Pops a value from the top of a V-memory table and stores it in a specified V-memory location. All other values in the V-memory table are shifted up each time a value is popped from the table.
- Add To Top of Table (ATT)**
Pushes a value from a specified V-memory location onto the top of a V-memory table. All other values in the V-memory table are shifted down each time a value is pushed onto the table.
- Table Shift Left (TSHFL)**
Shifts a specified number of bits to the left in a V-memory table.
- Table Shift Right (TSHFR)**
Shifts a specified number of bits to the right in a V-memory table.
- And Move (ANDMOV)**
Copies data from a table to the specified location, ANDING each word with the accumulator data as it is written.
- Or Move (ORMOV)**
Copies data from a table to the specified memory location, ORing each word with the accumulator data as it is written.
- Exclusive Or Move (XORMOV)**
Copies data from a table to the specified memory location, XORing each word with the accumulator data as it is written.
- Swap (SWAP)**
Exchanges the data in two tables of equal length.

Clock / Calendar Instructions

- Date (DATE)**
D2-250-1, D2-260 only. Use to set the date in the CPU.
- Time (TIME)**
D2-250-1, D2-260 only. Use to set the time in the CPU.

CPU Control Instructions

- No Operation (NOP)**
Inserts a no operation coil at specified program address.
- End (END)**
Marks the termination point for the normal program scan. An End instruction is required at the end of the main program body.
- Stop (STOP)**
Changes the operational mode of the CPU from Run to Program (Stop).
- Reset Watchdog Timer (RSTWT)**
D2-240, D2-250-1, D2-260 only. Resets the CPU watchdog timer.

DL205 Instruction Set

Program Control Instructions

Goto Label (GOTO) (LBL)

D2-240, D2-250-1, D2-260 only. Skips all instructions between the Goto and corresponding LBL instructions.

For/Next (FOR/NEXT)

D2-240, D2-250-1, D2-260 only. Executes the logic between the FOR and NEXT instructions a specified number of times.

Goto Subroutine (GTS/SBR/RT/RTC)

GTS, SBR, RT: D2-240, D2-250-1, D2-260 only.
RTC: D2-250-1, D2-260 only.

When a GTS instruction is executed the program jumps to the SBR (Subroutine). The subroutine is terminated with an RT instruction (unconditional return). When a return is executed, the program continues from the instruction after the calling GTS instruction. The RTC (Subroutine return conditional) instruction is used with an input contact to implement a conditional return from the subroutine.

Master Line Set/Master Line Reset (MLS/MLR)

Allows the program to control sections of ladder logic by forming a new power rail. The MLS marks the beginning of a power rail and the MLR marks the end of the power rail control.

Interrupt Instructions

Interrupt Routine/Interrupt Return/Interrupt Return Conditional (INT/IRT/IRTC)

INT, IRT: D2-240, D2-250-1, D2-260 only.
IRTC: D2-250-1, D2-260 only.

When a hardware or software interrupt occurs, the interrupt routine will be executed. The INT instruction is the beginning of the interrupt routine. The interrupt routine is terminated with an IRT of the interrupt routine. The in interrupt routine is terminated with an IRT instruction (unconditional interrupt return). When an interrupt return is reached the execution of the program continues from the instruction where the program execution was prior to the interrupt.

Enable Interrupt (ENI)

D2-240, D2-250-1, D2-260 only. Enables hardware and software interrupts to be acknowledged.

Disable Interrupt (DISI)

D2-240, D2-250-1, D2-260 only. Disables hardware and software interrupts from being acknowledged.

Intelligent I/O Instructions

Read from Intelligent Module (RD)

Reads a block of data from an intelligent I/O module into CPU's V-memory.

Write to Intelligent Module (WT)

Writes a block of data to an intelligent I/O module from a block of CPU's V-memory.

Message Instructions

Fault/Data Label (FAULT/DLBL)

FAULT: D2-240, D2-250-1, D2-260 only.
DLBL: All D2 CPUs.

Displays a V-memory value or a data label constant to the hand-held programmer or personal computer using DirectSOFT.

Numerical Constant/ASCII constant (NCON/ACON)

Stores constants in numerical or ASCII form for use with other instructions.

Print Message (PRINT)

D2-250-1, D2-260 only.
Prints the embedded text or text/data variable message to the specified communications port. Maximum message length is 255 words.

Network Instructions

Read from network (RX)

D2-240, D2-250-1, D2-260 only. Reads a block of data from another CPU on the network.

Write to network (WX)

D2-240, D2-250-1, D2-260 only. Writes a block of data from the master device to a slave device on the network.

Modbus Instructions (D2-260 only)

Modbus Read (MRS)

Used CPU port 2 to read a block of data from Modbus RTU devices on the network.

Modbus Write (MWX)

Writes a block of data from CPU port 2 to Modbus RTU devices on the network.

ASCII Instructions (D2-260 only)

ASCII IN (AIN)

Configures port 2 to read raw ASCII input strings.

ASCII Find (AFIND)

Searches ASCII strings in V-memory to find a specific portion of the string.

ASCII IN (AEX)

Extracts a specific portion from an ASCII string.

Compare V-memory (CMPV)

Compares two blocks of V-memory.

Swap Bytes (SWAPB)

Swaps V-memory bytes.

Print to V-memory (VPRINT)

Used to send pre-coded ASCII strings to a pre-defined V-memory address when enabled.

Print from V-memory (PRINTV)

Used to write raw ASCII string out of port 2 when enabled.

Drum Instructions (D2-250-1, D2-260 only)

Tuned Drum with Discrete Outputs (DRUM)

Time driven drum with up to 16 steps and 16 discrete output points. Output status is written to the appropriate output during each step. Specify a time base per count (in milliseconds). Each step can have a different number of counts to trigger the transition to the next step. Also define preset step as destination when reset occurs.

Time & Event Drum with Discrete Outputs (EDRUM)

Time and/or event driven drum with up to 16 steps and 16 discrete output points. Output status is written to the appropriate output during each step. Specify a time base per count (in milliseconds). Each step can have a different number of counts and an event to trigger the counting. Once the time has expired, a transition to the next step occurs. Also define preset step as destination when reset occurs.

Time and Event Drum with Discrete Outputs and Output Mask (MDRMD)

Time and/or event driven drum with up to 16 steps and 16 discrete output points. Actual output status is the result of a bit-by-bit AND between the output mask and bit mask in the step. Specify a time base per count (in milliseconds). Each step can have a different number of counts and an event to trigger the counting. Once the time has expired, a transition to the next step occurs. Also, define present step as destination when reset occurs.

Time and Event Drum with Word Output and Output Mask (MDRMW)

Time and/or event driven drum with up to 16 steps and a single V-memory output location. Actual output word is the result of a bit-by-bit AND between the word mask and the bit mask in the step. Specify a time base per count (in milliseconds). Each step can have a different number of counts and an event to trigger the counting. Once the time has expired, a transition to the next step occurs. Also define preset step as destination when reset occurs.

RLL^{PLUS} Programming Instructions

(D2-240, D2-250-1, D2-260 only)

Initial stage (ISG)

The initial stage instruction is used for a starting point for user application program. The ISG instruction will be active on power up and PROGRAM to RUN transitions.

Stage (SG)

Stage instructions are used to create structured programs. They are program segments which can be activated or deactivated with control logic.

Jump (JMP)

Normally open coil that deactivates the active stage and activates a specified stage when there is power flow to the coil.

Not Jump (NJMP)

Normally closed coil that deactivates the active stage and activates a specified stage when there is power flow to the coil.

Converge Stages (CV)

Converge stages are a group of stages that when all stages are active the associated converge jump(s). (CVJMP) will activate another stage(s). One scan after the CVJMP is executed, the converge stages will be deactivated.

Converge Jump (CVJMP)

Normally open coil that deactivates the active CV stages and activates a specified stage when there is power flow to the coil.

Block Call/Block/Block End (BCALL w/BLK and BEND)

BCALL is a normally open coil that activates a block of stages when there is power flow to the coil. BLK is the label which marks the beginning of a block of stages. Bend is a label used to mark the end of a block of stages.

Here are some of the IBox Instructions available

The IBox instructions are available when using a **D2-250-1 CPU with firmware version 4.60 or later, D2-260 CPU with firmware version 2.40 or later, and DirectSOFT5 or later.**

IBox Instructions - Analog Helper

Analog Input/Output Combo Module Pointer Setup (ANLGCMB)

Generates the logic to configure the pointer method for an analog input/output combination module.

Analog Input Module Pointer Setup (ANLGIN)

Generates the logic to configure the pointer method for an analog input module.

Analog Output Module Pointer Setup (ANLGOUT)

Generates the logic to configure the pointer method for an analog output module.

Analog Scale 12 Bit BCD to BCD (ANSCL)

Scales a 12 bit BCD analog value (0-4095 BCD) into BCD engineering units. Only works with unipolar unsigned raw values.

Analog Scale 12 Bit Binary to Binary (ANSCLB)

Scales a 12 bit binary analog value (0-4095 decimal) into Binary engineering units. Only works with unipolar unsigned raw values.

Filter Over Time - BCD (FILTER)

Performs a first-order filter on the Raw Data on a defined time interval (BCD).

Filter Over Time - Binary (FILTERB)

Perform a first-order filter on the Raw Data on a defined time interval (binary).

Hi/Low Alarm - BCD (HILOAL)

Monitors a BCD value V memory location and sets four possible alarm states, High-High, High, Low, and Low-Low.

Hi/Low Alarm - Binary (HILOALB)

Monitors a binary (decimal) value V memory location and sets four possible alarm states, High-High, High, Low, and Low-Low.

IBox Instructions - Discrete Helper

Off Delay Timer - (OFFDTMR)

Delays the "turning off" of the Output parameter by the specified Off Delay Time (in hundredths of a second).

On Delay Timer - (ONDTMR)

Delays the "turning on" of the Output parameter by the specified amount of time (in hundredths of a second).

One Shot - (ONESHOT)

Turns on the given bit output parameter for one scan on an OFF to ON transition.

Push On / Push Off Circuit (PONOFF)

Toggles an output state whenever its input power flow transitions from off to on. Also known as a "flip-flop" circuit.

IBox Instructions - Memory

Move Single Word (MOVEV)

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.

Move Double Word (MOVED)

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.

IBox Instructions - Math

BCD to Real with Implied Decimal Point (BCDTR)

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

Double BCD to Real with Implied Decimal Point (BCDTRD)

Converts the given 8 digit DWORD BCD value to a Real number, given an implied number of decimal points (K0-K8).

Math - BCD (MATHBCD)

Allows entry of complex mathematical expressions like in Visual Basic, Excel, or C++ to do complex calculations, nesting parentheses up to 4 levels deep. Every V-memory reference MUST be to a single word BCD formatted value.

Math - Binary (MATHBIN)

Allows entry of complex mathematical expressions like in Visual Basic, Excel, or C++ to do complex calculations, nesting parentheses up to 4 levels deep. Every V-memory reference MUST be to a single word binary formatted value.

Math - Real (MATHR)

Allows entry of complex mathematical expressions like in Visual Basic, Excel, or C++ to do complex calculations, nesting parentheses up to 4 levels deep. Every V-memory reference MUST be to fit into a double word Real formatted value.

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

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.

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

Converts the absolute value of the given Real number to an 8 digit DWORDBCD number, compensating for an implied number of decimal points (K0-K8) and performs rounding.

Square BCD (SQUARE)

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

Square Binary (SQUAREB)

Squares the given 16-bit WORD binary number and writes it as a 32-bit DWORD binary result.

Square Real (SQUARER)

Squares the given REAL DWORD number and writes it to a REAL DWORD result.

Sum BCD Numbers (SUMBCD)

Sums a list of consecutive 4-digit WORD BCD numbers into an 8-digit DWORD BCD result.

Sum Binary Numbers (SUMBIN)

Sums a list of consecutive 16-bit WORD binary numbers into a 32-bit DWORD binary result.

Sum Real Numbers (SUMR)

Sums a list of consecutive Real DWORD numbers into a Real DWORD result.

IBox Instructions - Communications

ECOM100 Configuration (ECOM100)

Defines the common information for a specific ECOM100 module which is used by the other ECOM100 IBoxes and resides at the top of the ladder/stage program. If using more than one ECOM100 in a PLC system, a different ECOM100 Configuration IBox must be used for each ECOM100 module that utilizes ECOM IBox instructions.

ECOM100 Disable DHCP (ECDHCPD)

Commands the ECOM100 to use its internal TCP/IP settings.

ECOM100 Enable DHCP (ECDHCP)

Commands the ECOM100 to obtain its TCP/IP settings from a DHCP server.

ECOM100 Query DHCP Setting (ECDHCPQ)

Determines if DHCP is enabled in the ECOM100.

ECOM100 Send E-mail (ECEMAIL)

Allows the ECOM100 to behave as an EMail client to send an SMTP request to the SMTP Server for sending the EMail messages to EMail addresses in the To: field and also to those listed in the Cc: list hard coded in the ECOM100. Messages are limited to 100 characters for the entire instruction.

ECOM100 Restore Default E-mail Setup (ECEMRDS)

Restores the original EMail Setup data stored in the ECOM100 back to the working copy based on the specified ECOM100#.

ECOM100 E-mail Setup (ECESUP)

Modifies the working copy of the EMail setup currently in the ECOM100 based on the specified ECOM100#. You may pick and choose any or all fields to be modified using this instruction.

ECOM100 IP Setup (ECIPSUP)

Configures the three TCP/IP parameters in the ECOM100: IP Address, Subnet Mask and Gateway Address.

ECOM100 Read Description (ECRDDES)

Reads the ECOM100's Description field up to the number of specified characters.

ECOM100 Read Gateway Address (ECRDGWA)

Reads the ECOM100's Gateway address and stores it in 4 consecutive V memory locations in decimal format.

ECOM100 Read IP Address (ECRDIP)

Reads the ECOM100's IP address and stores it 4 consecutive V memory locations in decimal format.

ECOM100 Read Module ID (ECRDMID)

Reads the ECOM100's binary (decimal) WORD sized Module ID and stores it in V memory.

ECOM100 Read Module Name (ECRDNAM)

Reads the ECOM100's Module Name up to the number of specified characters and stores it in V memory.

ECOM100 Read Subnet Mask (ECRDSNM)

Reads the ECOM100's Subnet Mask address and stores it 4 consecutive V memory locations in decimal format.

ECOM100 Write Description (ECWRDES)

Writes the specified Description to the ECOM100 module.

ECOM100 Write Gateway Address (ECWRGWA)

Writes the specified Gateway IP Address to the ECOM100 module.

ECOM100 Write IP Address (ECWRIP)

Writes the specified IP Address to the ECOM100 module.

ECOM100 Write Module ID (ECWRMID)

Writes the specified Module ID to the ECOM100 module.

ECOM100 Write Name (ECWRNAM)

Writes the specified Name to the ECOM100 module.

ECOM100 Write Subnet Mask (ECWRSNM)

Writes the specified Subnet Mask to the ECOM100 module.

ECOM100 RX Network Read (ECRX)

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.

ECOM100 WX Network Write (ECWX)

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.

NETCFG Network Configuration (NETCFG)

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.

Network RX Read (NETRX)

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.

Network WX Read (NETWX)

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.

IBox Instructions - Counter I/O

CTRIO Configuration (CTRIO)

Defines the common information for a specific CTRIO module which is used by the other CTRIO IBox instructions and resides at the top of the ladder/stage program. If using more than one CTRIO module in a PLC system, a different CTRIO Configuration IBox must be used for each CTRIO module that utilizes any CTRIO IBox instructions.

CTRIO Add Entry to End of Preset Table (CTRADPT)

Appends an entry to the end of a memory based Preset Table on a specific CTRIO Output resource. Will take more than 1 PLC scan to execute.

CTRIO Clear Preset Table (CTRCLRT)

Clears the RAM based Preset Table on a leading edge transition to this IBox. Will take more than 1 PLC scan to execute.

CTRIO Edit Preset Table Entry (CTREDPT)

Edits a single entry in a Preset Table on a specific CTRIO Output resource. Will take more than 1 PLC scan to execute.

CTRIO Edit Preset Table Entry and Reload (CTREDRL)

Performs dual operation to a CTRIO Output resource in one CTRIO command. Will take more than 1 PLC scan to execute.

CTRIO Initialize Preset Table (CTRINPT)

Creates a single entry Preset Table in memory but not as a file, on a specific CTRIO Output resource. Will take more than 1 PLC scan to execute.

CTRIO Initialize Preset Table on Reset (CTRINTR)

Configures the initial Preset Table to be automatically loaded whenever the Reset event occurs on a specific Output resource. Will take more than 1 PLC scan to execute.

CTRIO Load Profile (CTRLDPR)

Loads a CTRIO Profile File to a CTRIO Output resource on a leading edge transition to this IBox. Will take more than 1 PLC scan to execute.

CTRIO Read Error (CTRRDER)

Gets the decimal error code value from the CTRIO module and places it into the specified Error Code register. Since the Error Code in the CTRIO is only maintained until another CTRIO command is given, this instruction must be used immediately after the CTRIO IBox that reports an error via its Error bit parameter.

CTRIO Run to Limit Mode (CTRRLTM)

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 operate. Will take more than 1 PLC scan to execute.

CTRIO Run to Position Mode (CTRRTPM)

Loads the Run to Position command and given parameters on a specific Output resource. Will take more than 1 PLC scan to execute.

CTRIO Velocity Mode (CTRVELO)

Loads the Velocity command and given parameters on a specific Output resource. Will take more than 1 PLC scan to execute.

CTRIO Write File to ROM (CTRWFTR)

Writes the runtime changes made to a loaded CTRIO Preset Table back to Flash ROM. Will take more than 1 PLC scan to execute.