Chapter Contents

5. Applied Instructions

FX1S	FX1N	FX2N	FX2NC

Applied Instructions are the 'specialist' instructions of the FX family of PLC's. They allow the user to perform complex data manipulations, mathematical operations while still being very easy to program and monitor. Each applied instruction has unique mnemonics and special function numbers. Each applied instruction will be expressed using a table similar to that shown below:

Mnemonic	Function	Operands	Program steps
		D	
CJ FNC 00 (Conditional Jump)	A method of jumping to an identified pointer position	Valid pointers from the range 0 to 63	CJ,CJP:3steps Jump pointer P☆☆:1 step

The table will be found at the beginning of each new instruction description. The area identified as 'Operands' will list the various devices (operands) that can be used with the instruction. Various identification letters will be used to associate each operand with its function, i.e. D- destination, S- source, n, m- number of elements. Additional numeric suffixes will be attached if there are more than one operand with the same function.

Not all instructions and conditions apply to all PLC's. Applicable CPU's are identified by the boxes in the top right hand corner of the page. For more detailed instruction variations a second indicator box is used to identify the availability of pulse, single (16 bit) word and double (32 bit) word format and to show any flags that are set by the instruction.

16 BIT OPERATION	32 BIT OPERATION	PULSE-P

No modification of the instruction mnemonic is required for 16 bit operation. However, pulse operation requires a 'P' to be added directly after the mnemonic while 32 bit operation requires a 'D' to be added before the mnemonic. This means that if an instruction was being used with both pulse and 32 bit operation it would look like..... D☆☆☆P where ☆☆☆ was the basic mnemonic.

The 'pulse' function allows the associated instruction to be activated on the rising edge of the control input. The instruction is driven ON for the duration of one program scan.
Thereafter, while the control input remains ON, the associated instruction is not active.
To re-execute the instruction the control input must be turned from OFF to ON again.
The FLAGS section identifies any flags that are used by the instruction. Details about the function of the flag are explained in the instructions text.

- For instructions that operate continuously, i.e. on every scan of the program the instruction will operate and provide a new, different result, the following identification symbol will be used '✈' to represent a high speed changing state. Typical instructions covered by this situation have a strong incremental, indexable element to their operation.

- In most cases the operands of applied instructions can be indexed by a users program. For those operands which **cannot** be indexed, the symbol '✉' has been used to signify an operand as being 'fixed' after it has been written.

- Certain instructions utilize additional data registers and/or status flags for example a math function such as ADD (FNC 20) can identify a zero result, borrow and carry conditions by using preset auxiliary relays, M8020 to M8021 respectively.

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.1 Program Flow-Functions 00 to 09

FX1S | FX1N | FX2N | FX2NC

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

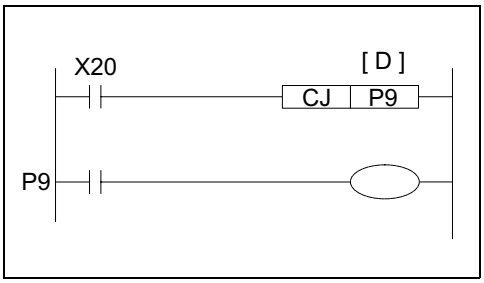
✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.1.1 CJ (FNC 00)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		D	
CJ FNC 00 (Conditional Jump)	Jumps to the identified pointer position	Valid pointers from the range 0 to 63	CJ, CJP:3steps Jump pointer P☆☆: 1 step

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

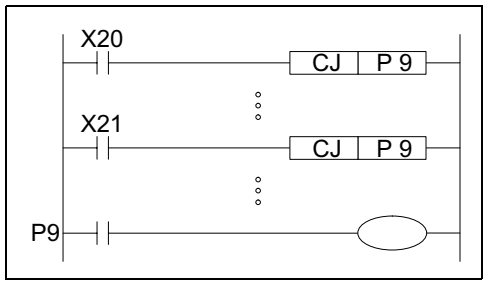
When the CJ instruction is active it forces the program to jump to an identified program marker. While the jump takes place the intervening pro-gram steps are skipped. This means they are not processed in any way. The resulting effect is to speed up the programs operational scan time.

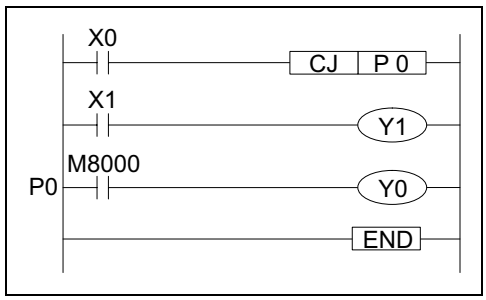
Points to note:

a) Many CJ statements can reference a single pointer.

b) Each pointer must have a unique number. Using pointer P63 is equivalent to jumping to the END instruction.

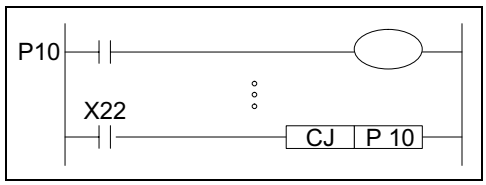
c) Any program area which is skipped, will not update output statuses even if the input devices change. For example, the program opposite shows a situation which loads X1 to drive Y1. Assuming X1 is ON and the CJ instruction is activated the load X1, out Y1 is skipped. Now even if X1 is turned OFF Y1 will remain ON while the CJ instruction forces the program to skip to the pointer P0. The reverse situation will also apply, i.e. if X1 is OFF to begin with and the CJ instruction is driven, Y1 will not be turned ON if X1 is turned ON. Once the CJ instruction is deactivated X1 will drive Y1 in the normal manner. This situation applies to all types of outputs, e.g. SET, RST, OUT, Y, M & S devices etc.





d) The CJ instruction can jump to any point within the main program body or after an FEND instruction

e) A CJ instruction can be used to Jump forwards through a program, i.e. towards the END instruction OR it can jump backwards towards step 0. If a backwards jump is used care must be taken not to overrun the watchdog timer setting otherwise the PLC will enter an error situation. For more information on the watchdog timer please see page 5-12.



f) Unconditional jumps can be entered by using special auxiliary coils such as M8000. In this situation while the PLC is in RUN the program will ALWAYS execute the CJ instruction in an unconditional manner.

IMPORTANT:

- Timers and counters will freeze their current values if they are skipped by a CJ instruction. For example if Y1 in the previous program (see point c) was replaced by T0 K100 and the CJ instruction was driven, the contents of T0 would not change/increase until the CJ instruction is no longer driven, i.e. the current timer value would freeze.
  High speed counters are the only exception to this situation as they are processed independently of the main program.

Using applied instructions:

- Applied instructions are also skipped if they are programmed between the CJ instruction and the destination pointer. However, The PLSY (FNC 57) and PWM (FNC 58) instructions will operate continuously if they were active before the CJ instruction was driven, otherwise they will be processed, i.e. skipped, as standard applied instructions.

Details of using CJ with other program flow instructions

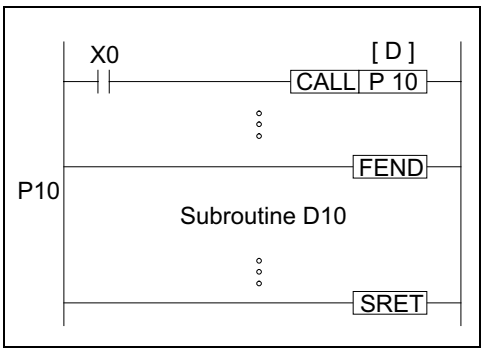
- Further details can be found on pages 7-12 and 7-13 about the combined use of different program flow techniques (such as master control, MC etc).

5.1.2 CALL (FNC 01)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		D	
CALL FNC 01 (Call sub-routine)	Executes the subroutine program starting at the identified pointer position	Valid pointers from the range 0 to 62 Nest levels: 5 including the initial CALL	CALL, CALLP: 3 step Subroutine pointer P☆☆: 1 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

When the CALL instruction is active it forces the program to run the subroutine associated with the called pointer (area identified as subroutine P10). A CALL instruction must be used in conjunction with FEND (FNC 06) and SRET (FNC 02) instructions. The program jumps to the subroutine pointer (located after an FEND instruction) and processes the contents until an SRET instruction is encountered. This forces the program flow back to the line of ladder logic immediately following the original CALL instruction.

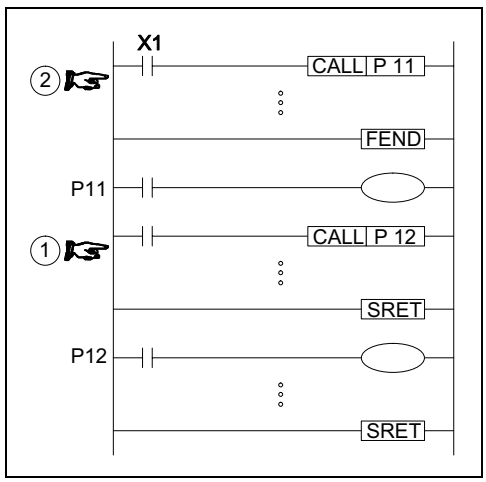
Points to note:

a) Many CALL statements can reference a single subroutine.

b) Each subroutine must have a unique pointer number. Subroutine pointers can be selected from the range P0 to P62. Subroutine pointers and the pointers used for CJ (FNC 00) instructions are NOT allowed to coincide.

c) Subroutines are not normally processed as they occur after an FEND instruction. When they are called, care should be taken not to overrun the watchdog timer setting. For more information on watchdog timers please see page 5-12.

d) Subroutines can be nested for 5 levels including the initial CALL instruction. As an example the program shown opposite shows a 2 level nest. When X1 is activated the program calls subroutine P11. Within this subroutine is a CALL to a second subroutine P12. When both subroutines P11 and P12 are active simultaneously, they are said to be nested. Once subroutine P12 reaches its SRET instruction it returns the program control to the program step immediately following its original CALL (see ①). P11 then completes its operation, and once its SRET instruction is processed the program returns once again to the step following the CALL P11 statement (see ②).



Special subroutine timers:

- Because of the chance of intermittent use of the subroutines, if timed functions are required the timers used must be selected from the range T192 to T199 and T246 to T249.

Details of using CALL with other program flow instructions

- Further details can be found on pages 7-12 and 7-13 about the combined use of different program flow techniques (such as master control, MC etc).

5.1.3 SRET (FNC 02)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		D	
SRET FNC 02 (Subroutine return)	Returns operation from a subroutine program	N/A Automatically returns to the step immediately following the CALL instruction which activated the subroutine.	SRET: 1 step

16 BIT OPERATION	32 BIT OPERATION	PULSE-P

Operation:

SRET signifies the end of the current subroutine and returns the program flow to the step immediately following the CALL instruction which activated the closing subroutine.

Points to note:

a) SRET can only be used with the CALL instruction.

b) SRET is always programmed after an FEND instruction - please see the CALL (FNC 01) instruction for more details.

5.1.4 IRET, EI, DI (FNC 03, 04, 05)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		D	
IRET FNC 03 (Interrupt return)	Forces the program to return from the active interrupt routine	N/A Automatically returns to the main program step which was being processed at the time of the interrupt call.	IRET: 1 step
EI FNC 04 (Enable interrupts)	Enables interrupt inputs to be processed	N/A Any interrupt input being activated after an EI instruction and before FEND or DI instructions will be processed immediately unless it has been specifically disabled.	EI: 1 step
DI FNC 05 (Disable interrupts)	Disables the processing of interrupt routines	N/A Any interrupt input being activated after a DI instruction and before an EI instruction will be stored until the next sequential EI instruction is processed.	DI: 1 step
I (Interrupt pointer)	Identifies the beginning of an interrupt routine	A 3 digit numeric code relating to the interrupt type and operation.	I☆☆☆: 1 step

General description of an interrupt routine:

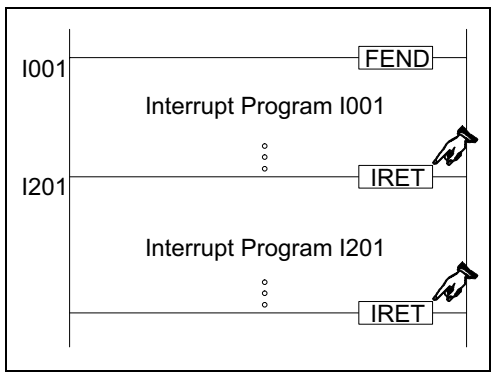
An interrupt routine is a section of program which is, when triggered, operated immediately interrupting the main program flow. Once the interrupt has been processed the main program flow continues from where it was, just before the interrupt originally occurred.

Operation:

Interrupts are triggered by different input conditions, sometimes a direct input such as X0 is used other times a timed interval e.g. 30 msec can be used. The availability of different interrupt types and the number operational points for each PLC type are detailed on page 4-12, Interrupt Pointers. To program and operate interrupt routines requires up to 3 dedicated instructions (those detailed in this section) and an interrupt pointer.

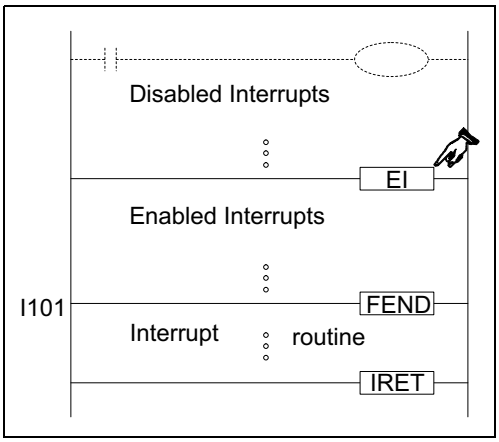
Defining an interrupt routine:

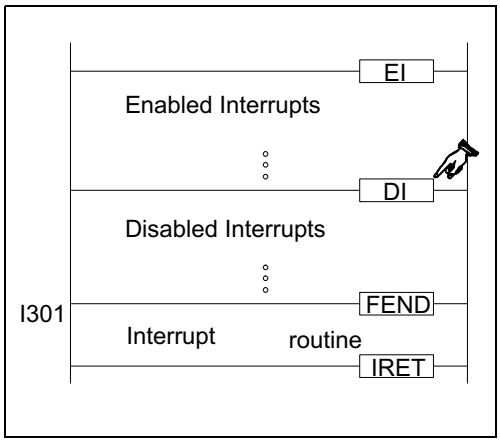
An interrupt routine is specified between its own unique interrupt pointer and the first occurrence of an IRET instruction.
Interrupt routines are ALWAYS programmed after an FEND instruction.
The IRET instruction may only be used within interrupt routines.



Controlling interrupt operations:

The PLC has a default status of disabling interrupt operation. The EI instruction must be used to activate the interrupt facilities. All interrupts which physically occur during the program scan period from the EI instruction until the FEND or DI instructions will have their associated interrupt routines run. If these interrupts are triggered outside of the enclosed range (EI-FEND or EI-DI, see diagram below) they will be stored until the EI instruction is processed on the following scan. At this point the interrupt routine will be run.





If an individual interrupt is to be disabled its associated special M coil must be driven ON. While this coil is ON the interrupt routine will not be activated. For details about the disabling M coils see the PLC device tables in chapter 8.

Nesting interrupts:

Interrupts may be nested for two levels. This means that an interrupt may be interrupted during its operation. However, to achieve this, the interrupt routine which may be further interrupted must contain the EI and DI instructions; otherwise as under normal operation, when an interrupt routine is activated all other interrupts are disabled.

Simultaneously occurring interrupts:

If more than one interrupt occurs sequentially, priority is given to the interrupt occurring first. If two or more interrupts occur simultaneously, the interrupt routine with the lower pointer number is given the higher priority.

Using general timers within interrupt routines:

FX PLC's have a range of special timers which can be used within interrupt routines. For more information please see page 4-18, Timers Used in Interrupt and 'CALL' Subroutines.

Input trigger signals - pulse duration:

Interrupt routines which are triggered directly by interrupt inputs, such as X0 etc., require a signal duration of approximately 200µsec, i.e. the input pulse width is equal or greater than 200µsec. When this type of interrupt is selected, the hardware input filters are automatically reset to 50µsec. (under normal operating circumstances the input filters are set to 10msec.).

Pulse catch function:

Direct high speed inputs can be used to 'catch' short pulsed signals. When a pulse is received at an input a corresponding special M coil is set ON. This allows the 'captured' pulse to be used to trigger further actions, even if the original signal is now OFF. FX_{1S}, FX_{1N}, FX_{2N} and FX_{2NC} units require the EI instruction (FNC 04) to activate pulse catch for inputs X0 through X5, with M8170 to M8175 indicating the caught pulse. Note that, if an input device is being used for another high speed function, then the pulse catch for that device is disabled.

5.1.5 FEND (FNC 06)

FX1S	FX1N	FX2N	FX2NC

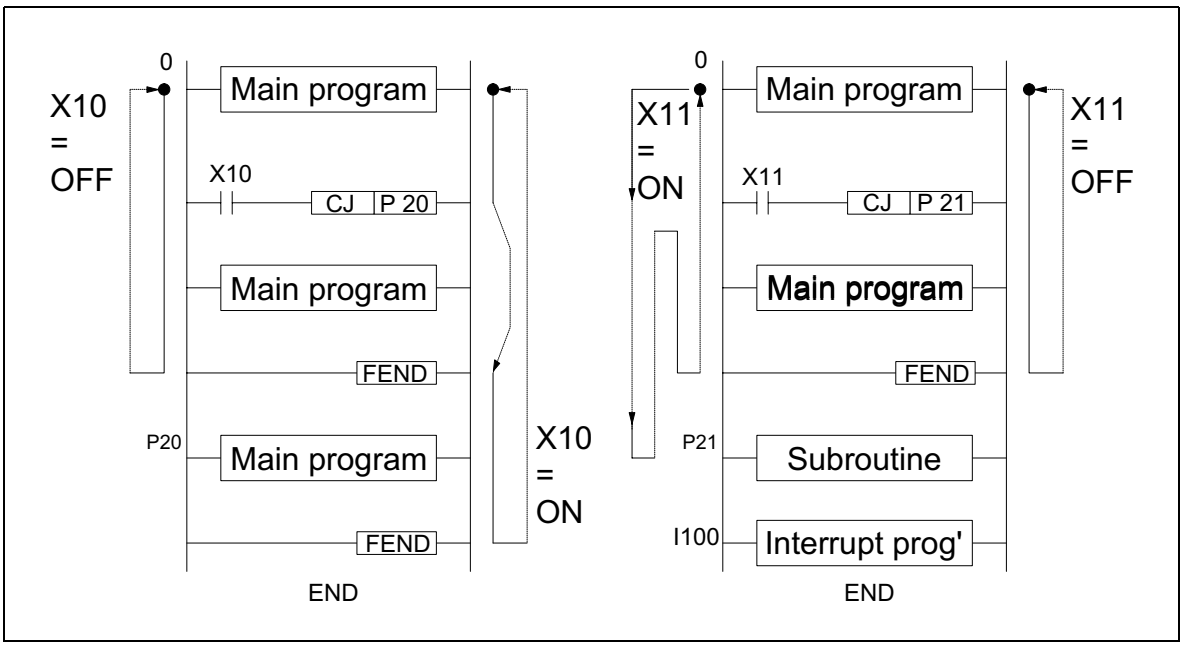
Mnemonic	Function	Operands	Program steps
		D	
FEND FNC 06 (First end)	Used to indicate the end of the main program block	N/A Note: Can be used with CJ (FNC 00), CALL (FNC 01) and interrupt routines	FEND: 1 step

Operation:

An FEND instruction indicates the first end of a main program and the start of the program area to be used for subroutines. Under normal operating circumstances the FEND instruction performs a similar action to the END instruction, i.e. output processing, input processing and watchdog timer refresh are all carried out on execution.

Points to note:

a) The FEND instruction is commonly used with CJ-P-FEND, CALL-P-SRET and I-IRET program constructions (P refers to program pointer, I refers to interrupt pointer).
Both CALL pointers/subroutines and interrupt pointers (I) subroutines are ALWAYS programmed after an FEND instruction, i.e. these program features NEVER appear in the body of a main program.



b) Multiple occurrences of FEND instructions can be used to separate different subroutines (see diagram above).

c) The program flow constructions are NOT allowed to be split by an FEND instruction.

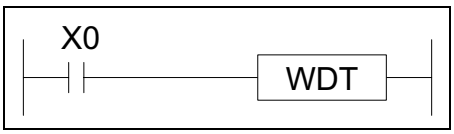
d) FEND can never be used after an END instruction.

5.1.6 WDT (FNC 07)

FX1S	FX1N	FX2N	FX2NC

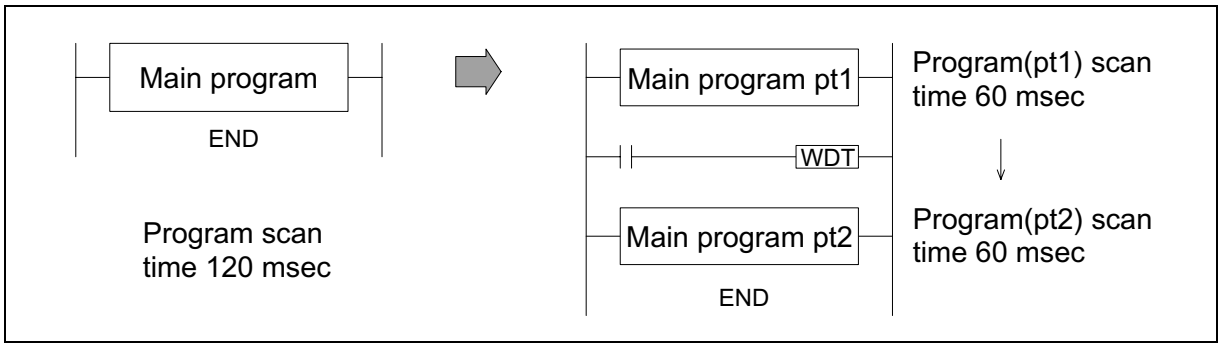
Mnemonic	Function	Operands	Program steps
		D	
WDT FNC 07 (Watch dog timer refresh)	Used to refresh the watch dog timer during a program scan	N/A Can be driven at any time within the main program body	WDT, WDTP: 1 step

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

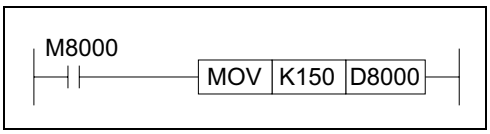
The WDT instruction refreshes the PLC's watchdog timer. The watchdog timer checks that the program scan (operation) time does not exceed an arbitrary time limit. It is assumed that if this time limit is exceeded there is an error at some point. The PLC will then cease operation to prevent any further errors from occurring. By causing the watchdog timer to refresh (driving the WDT instruction) the usable scan (program operation) time is effectively increased.



Points to note:

a) When the WDT instruction is used it will operate on every program scan so long as its input condition has been made.
To force the WDT instruction to operate for only ONE scan requires the user to program some form of interlock.

b) The watchdog timer has a default setting of 200 msec. This time limit may be customized to a users own requirement by editing the contents of data register D8000, the watchdog timer register.

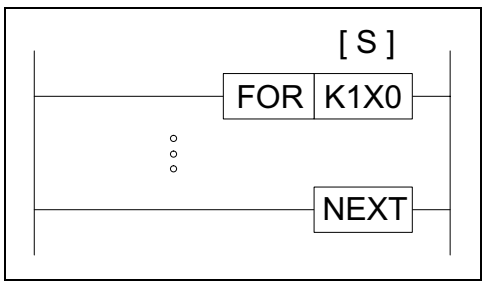


5.1.7 FOR, NEXT (FNC 08, 09)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		S	
FOR FNC 08 (Start of a FOR-NEXT loop)	Identifies the start position and the number of repeats for the loop	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	FOR: 3 step
NEXT FNC 09 (End of a FOR-NEXT loop)	Identifies the end position for the loop	N/A Note: The FOR-NEXT loop can be nested for 5 levels, i.e. 5 FOR-NEXT loops are programmed within each other.	NEXT: 1 step

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The FOR and NEXT instructions allow the specification of an area of program, i.e. the program enclosed by the instructions, which is to be repeated S number of times.

Points to note:

a) The FOR instruction operates in a 16 bit mode hence, the value of the operand S may be within the range of 1 to 32,767. If a number between the range -32,768 and 0 (zero) is specified it is automatically replaced by the value 1, i.e. the FOR-NEXT loop would execute once.

b) The NEXT instruction has NO operand.

c) The FOR-NEXT instructions must be programmed as a pair e.g. for every FOR instruction there **MUST** be an associated NEXT instruction. The same applies to the NEXT instructions, there **MUST** be an associated FOR instruction. The FOR-NEXT instructions must also be programmed in the correct order. This means that programming a loop as a NEXT-FOR (the paired NEXT instruction proceeds the associated FOR instruction) is **NOT** allowed.
Inserting an FEND instruction between the FOR-NEXT instructions, i.e. FOR-FEND- NEXT, is NOT allowed. This would have the same effect as programming a FOR without a NEXT instruction, followed by the FEND instruction and a loop with a NEXT and no associated FOR instruction.

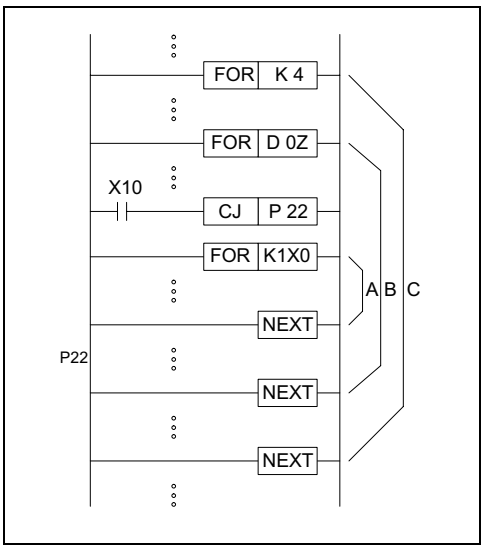
d) A FOR-NEXT loop operates for its set number of times **before** the main program is allowed to finish the current program scan.

e) When using FOR-NEXT loops care should be taken not the exceed the PLC's watchdog timer setting. The use of the WDT instruction and/or increasing the watchdog timer value is recommended.

Nested FOR-NEXT loops:

FOR-NEXT instructions can be nested for 5 levels. This means that 5 FOR-NEXT loops can be sequentially programmed within each other.
In the example a 3 level nest has been programmed. As each new FOR-NEXT nest level is encountered the number of times that loop is repeated is increased by the multiplication of all of the surrounding/previous loops.

For example, loop C operates 4 times. But within this loop there is a nested loop, B. For every completed cycle of loop C, loop B will be completely executed, i.e. it will loop D0Z times. This again applies between loops B and A.



The total number of times that loop A will operate for ONE scan of the program will equal;

1) The number of loop A operations multiplied by

2) The number of loop B operations multiplied by

3) The number of loop C operations

If values were associated to loops A, B and C, e.g. 7, 6 and 4 respectively, the following number of operations would take place in ONE program scan:

Number of loop C operations = 4 times
Number of loop B operations = 24 times (C $\times$ B, 4 $\times$ 6)
Number of loop A operations = 168 times (C $\times$ B $\times$ A, 4 $\times$ 6 $\times$ 7)

Note:

The use of the CJ programming feature, causing the jump to P22 allows the ‘selection’ of which loop will be processed and when, i.e. if X10 was switched ON, loop A would no longer operate.

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.2 Move And Compare - Functions 10 to 19

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

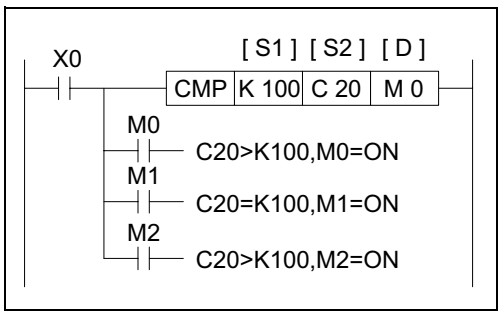
⊠ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.2.1 CMP (FNC 10)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S_1	S_2	D	
CMP FNC 10 (Compare)	Compares two data values - results of <, = and > are given.	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		Y, M, S Note: 3 consecutive devices are used.	CMP, CMPP: 7 steps DCMP, DCMPP: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The data of S_1 is compared to the data of S_2. The result is indicated by 3 bit devices specified from the head address entered as D. The bit devices indicate:
S_2 is less than S_1 - bit device D is ON
S_2 is equal to S_1 - bit device D_{+1} is ON
S_2 is greater than S_1 - bit device D_{+2} is ON

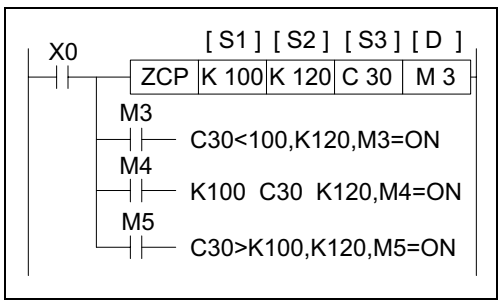
Note: The destination (D) device statuses will be kept even if the CMP instruction is deactivated. Full algebraic comparisons are used, i.e. -10 is smaller than +2 etc.

5.2.2 ZCP (FNC 11)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S_1	S_2	S_3	D	
ZCP FNC 11 (Zone compare)	Compares a data value against a data range - results of <, = and > are given.	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z Note: S_1 should be less than S_2			Y, M, S Note: 3 consecutive devices are used.	ZCP,Z CPP: 9 steps DZCP, DZCPP: 17 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The operation is the same as the CMP instruction except a single data value (S_3) is compared against a data range (S_1-S_2).

S_3 is less than $S_1$and S_2- bit device D is ON

S_3 is equal to or between S_1 and S_2 - bit device D_{+1} is ON

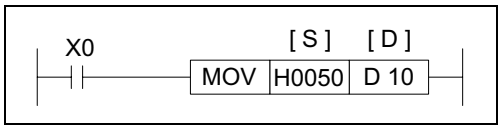
S_3 is greater than both S_1 and S_2 - bit device D_{+2} is ON

5.2.3 MOV (FNC 12)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
MOV FNC 12 (Move)	Moves data from one storage area to a new storage area	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	MOV, MOVP: 5 steps DMOV, DMOVP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Zero M8020



Operation:

The contents of the source device (S) is copied to the destination (D) device when the control input is active. If the MOV instruction is not driven, no operation takes place.

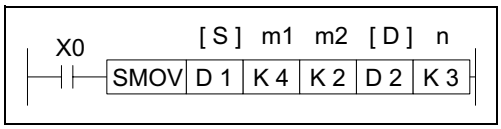
Note: This instruction has a special programming technique which allows it to mimic the operation of newer applied instructions when used with older programming tools. See page 1-5 for more details.

5.2.4 SMOV (FNC 13)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands					Program steps
		m1	m2	n	S	D	
SMOV FNC 13 (Shift move)	Takes elements of an existing 4 digit decimal number and inserts them into a new 4 digit number	K, H Note: available range 1 to 4. ⊠			K, H, KnX, KnY, KnM, KnS, T,C,D,V,Z	K, H, KnY, KnM, KnS, T,C,D,V,Z	SMOV, SMOVP: 11 steps
					Range 0 to 9,999 (decimal) or 0 to 9,999 (BCD) when M8168 is used - see note opposite		

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



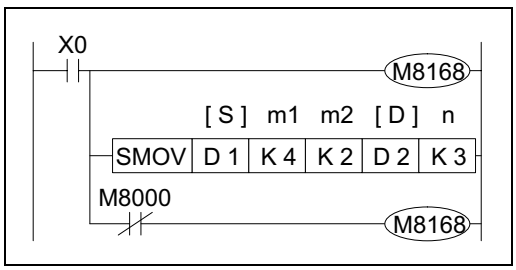
Operation 1:

This instruction copies a specified number of digits from a 4 digit decimal source (S) and places them at a specified location within a destination (D) number (also a 4 digit decimal). The existing data in the destination is overwritten.

Key:
m_1 - The source position of the 1st digit to be moved
m_2 - The number of source digits to be moved
n- The destination position for the first digit
Note: The selected destination must NOT be smaller than the quantity of source data.
Digit positions are referenced by number: 1= units, 2= tens, 3= hundreds, 4=thousands.

FX1S	FX1N	FX2N	FX2NC

Operation 2: (Applicable units, FX2N and FX2NC). This modification of the SMOV operation allows BCD numbers to be manipulated in exactly the same way as the 'normal' SMOV manipulates decimal numbers, i.e. This instruction copies a specified number of digits from a 4 digit BCD source (S) and places them at a specified location within a destination (D) number (also a 4 digit BCD number).



To select the BCD mode the SMOV instruction is coupled with special M coil M8168 which is driven ON. Please remember that this is a 'mode' setting operation and will be active, i.e. all SMOV instructions will operate in BCD format until the mode is reset, i.e. M8168 is forced OFF.

General note:

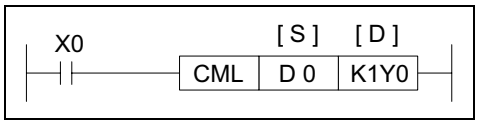
For more information about 'decimal' and 'Binary Coded Decimal' (BCD) numbers please see the section titled 'Interpreting Word Data' on page 4-42 for more details.

5.2.5 CML (FNC 14)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
CML FNC 14 (Compliment)	Copies and inverts the source bit pattern to a specified destination	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	CML,CMLP: 5 steps DCML, DCMLP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

A copy of each data bit within the source device (S) is inverted and then moved to a designated destination (D).

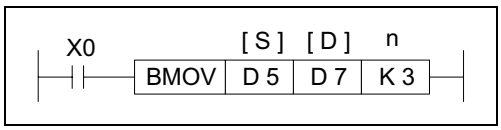
This means each occurrence of a ‘1’ in the source data will become a ‘0’ in the destination data while each source digit which is ‘0’ will become a ‘1’. If the destination area is smaller than the source data then only the directly mapping bit devices will be processed.

5.2.6 BMOV (FNC 15)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D	n	
BMOV FNC 15 (Block move)	Copies a specified block of multiple data elements to a new destination	KnX, KnY, KnM, KnS, T,C,D, V, Z (RAM) File registers,	KnY, KnM, KnS, T, C, D, V, Z (RAM) File registers, see note d)	K, H D (FX2C, FX2N only) ☒ Note: n≤ 512	BMOV, BMOVP: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

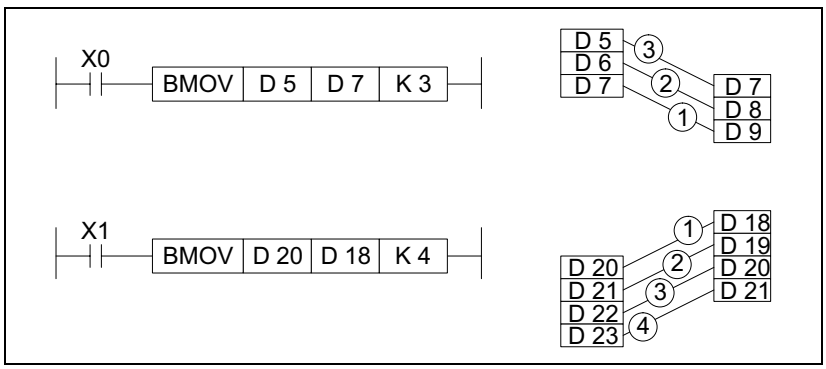
A quantity of consecutively occurring data elements can be copied to a new destination. The source data is identified as a device head address (S) and a quantity of consecutive data elements (n). This is moved to the destination device (D) for the same number of elements (n).

Points to note:

a) If the quantity of source devices (n) exceeds the actual number of available source devices, then only those devices which fall in the available range will be used.

b) If the number of source devices exceeds the available space at the destination location, then only the available destination devices will be written to.

c) The BMOV instruction has a built in automatic feature to prevent overwriting errors from occurring when the source (S - n) and destination (D -n) data ranges coincide. This is clearly identified in the following diagram:
(Note: The numbered arrows indicate the order in which the BMOV is processed)



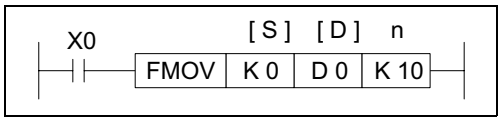
d) Using file registers as the destination devices [D]may be performed on all units.

5.2.7 FMOV (FNC 16)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D	n	
FMOV FNC 16 (Fill move)	Copies a single data device to a range of destination devices	KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	K, H ⊠ Note:n≤ 512	FMOV,FMOVP:7 steps DFMOV,DFMOVP: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The data stored in the source device (S) is copied to every device within the destination range. The range is specified by a device head address (D) and a quantity of consecutive elements (n). If the specified number of destination devices (n) exceeds the available space at the destination location, then only the available destination devices will be written to.

Note: This instruction has a special programming technique which allows it to mimic the operation of newer applied instructions when used with older programming tools. See page 1-5 for more details.

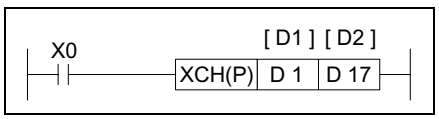
5.2.8 XCH (FNC 17)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		D1	D2	
XCH FNC 17 (Exchange) ✈	Data in the designated devices is exchanged	KnY, KnM, KnS, T, C, D, V, Z Note: when using the byte XCH (i.e.M8160 is ON) D1 and D2 must be the same device otherwise a program error will occur and M8067 will be turned ON		XCH,XCHP: 5 steps DXCH, DXCHP: 9 steps

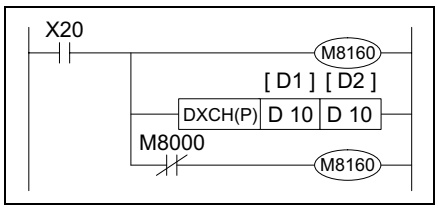
16 BIT OPERATION	32 BIT OPERATION	PULSE-P

Operation 1: The contents of the two destination devices D1 and D2 are swapped, i.e. the complete word devices are exchanged. Ex.



Data register	Before XCH	After XCH
D1	20	530
D17	530	20

Operation 2: This function is equivalent to FNC 147 SWAP The bytes within each word of the designated devices D1 are exchanged when 'byte mode flag' M8160 is ON. Please note that the mode will remain active until it is reset, i.e. M8160 is forced OFF. Ex.



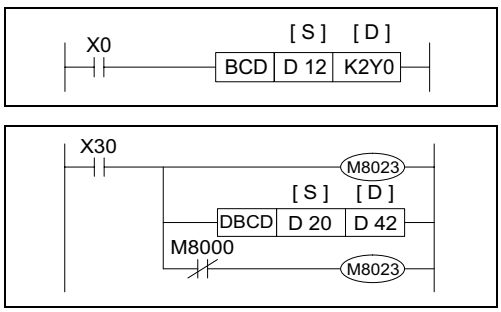
Values are in Hex for clarity		Before DXCH	After DXCH
D10	Byte 1	1FH	8BH
	Byte 1	8BH	1FH
D11	Byte 1	C4H	35H
	Byte 1	35H	C4H

5.2.9 BCD (FNC18)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
BCD FNC 18 (Binary coded decimal)	Converts binary numbers to BCD equivalents / Converts floating point data to scientific format	KnX,KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	BCD, BCDP: 5 steps DBCD, DBCDP: 9 steps
		When using M8023 to convert data to scientific format, only double word (32 bit) data registers (D) may be used. See page 4-46 for more details regarding floating point format.		

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



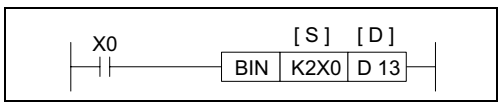
Operation: (Applicable to all units)
The binary source data (S) is converted into an equivalent BCD number and stored at the destination device (D). If the converted BCD number exceeds the operational ranges of 0 to 9,999 (16 bit operation) and 0 to 99,999,999 (32 bit operation) an error will occur.This instruction can be used to output data directly to a seven segment display.

5.2.10 BIN (FNC 19)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
BIN FNC 19 (Binary)	Converts BCD numbers to their binary equivalent / Converts scientific format data to floating point format	KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	BIN, BINP: 5 steps DBIN, DBINP: 9 steps
		When using M8023 to convert data to floating point format, only double word (32 bit) data registers (D) may be used. See page 4-46 for more details regarding floating point format.		

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation: (Applicable to all units)
The BCD source data (S) is converted into an equivalent binary number and stored at the destination device (D). If the source data is not provided in a BCD format an error will occur. This instruction can be used to read in data directly from thumbwheel switches.

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.3 Arithmetic And Logical Operations - Functions 20 to 29

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

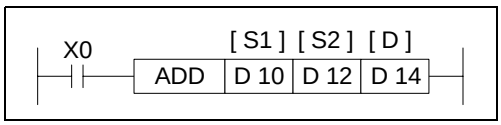
✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.3.1 ADD (FNC 20)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
ADD FNC 20 (Addition)	The value of the two source devices is added and the result stored in the destination device	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z	ADD, ADDP: 7 steps
		When using M8023 to add floating point data, only double word (32 bit) data registers (D) or constants (K/H) may be used. See page 4-46 for more details regarding floating point format.			DADD, DADDP: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation: (Applicable to all units)

The data contained within the source devices (S_1,S_2) is combined and the total is stored at the specified destination device (D).

Points to note:

a) All calculations are algebraically processed, i.e. 5 + (-8)= -3.

b) The same device may be used as a source (S_1 or S_2) and as the destination (D). If this is the case then the ADD instruction would actually operate continuously. This means on every scan the instruction would add the result of the last scan to the second source device. To prevent this from happening the pulse modifier should be used or an interlock should be programmed.

c) If the result of a calculation is “0" then a special auxiliary flag, M8020 is set ON.

d) If the result of an operation exceeds 32,767 (16 bit limit) or 2,147,483,647 (32 bit limit) the carry flag, M8022 is set ON. If the result of an operation exceeds -32,768 or -2,147,483,648 the borrow flag, M8021 is set ON. When a result exceeds either of the number limits, the appropriate flag is set ON (M8021 or M8022) and a portion of the carry/borrow is stored in the destination device. The mathematical sign of this stored data is reflective of the number limit which has been exceeded, i.e. when -32,768 is exceeded negative numbers are stored in the destination device but if 32,767 was exceeded positive numbers would be stored at D.

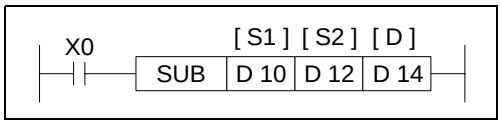
e) If the destination location is smaller than the obtained result, then only the portion of the result which directly maps to the destination area will be written, i.e if 25 (decimal) was the result, and it was to be stored at K1Y4 then only Y4 and Y7 would be active. In binary terms this is equivalent to a decimal value of 9 a long way short of the real result of 25!

5.3.2 SUB (FNC 21)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
SUB FNC 21 (Subtract)	One source device is subtracted from the other - the result is stored in the destination device	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z	SUB, SUBP: 7steps DSUB, DSUBP: 13 steps
		When using M8023 to subtract floating point data, only double word (32 bit) data registers (D) or constants (K/H) may be used. See page 4-46 for more details regarding floating point format.			

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Zero M8020 Borrow M8021 Carry M8022



Operation: (Applicable to all units)

The data contained within the source device, S_2 is subtracted from the contents of source device S_1. The result or remainder of this calculation is stored in the destination device D.

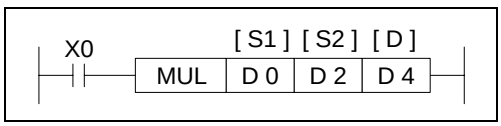
Note: the ‘Points to note’, under the ADD instruction (previous page) can also be similarly applied to the subtract instruction.

5.3.3 MUL (FNC 22)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
MUL FNC 22 (Multiplica-tion)	Multiplies the two source devices together the result is stored in the destination device	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z See page 4-46 for more details regarding floating point format.		KnY,KnM,KnS, T, C, D, Z(V) Note: Z(V) may **NOT** be used for 32 bit operation	MUL, MULP: 7steps DMUL, DMULP: 13 steps
		When using M8023 to subtract floating point data, only double word (32 bit) data registers (D) or constants (K/H) may be used.			

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation: (Applicable to all units)
The contents of the two source devices (S1, S2) are multiplied together and the result is stored at the destination device (D). Note the normal rules of algebra apply.

Points to note:

a) When operating the MUL instruction in 16bit mode, two 16 bit data sources are multiplied together. They produce a 32 bit result. The device identified as the destination address is the lower of the two devices used to store the 32 bit result. Using the above example with some test data:
5 (D0) $\times$ 7 (D2) = 35 - The value 35 is stored in (D4, D5) as a single 32 bit word.

b) When operating the MUL instruction in 32 bit mode, two 32 bit data sources are multiplied together. They produce a 64 bit result. The device identified as the destination address is the lower of the four devices used to store the 64 bit result.

c) If the location of the destination device is smaller than the obtained result, then only the portion of the result which directly maps to the destination area will be written, i.e if a result of 72 (decimal) is to be stored at K1Y4 then only Y7 would be active. In binary terms this is equivalent to a decimal value of 8, a long way short of the real result of 72!

Viewing 64 bit numbers

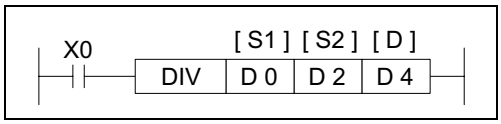
- It is currently impossible to monitor the contents of a 64 bit result. However, the result can be monitored in two smaller,32 bit, blocks, i.e. a 64 bit result is made up of the following parts: (upper 32 bits) $\times 2^{32}$ + (lower 32 bits).

5.3.4 DIV (FNC 23)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
DIV FNC 23 (Division)	Divides one source value by another the result is stored in the destination device	K, H, KnX, KnY, KnM, KnS,T, C, D, V, Z		KnY, KnM, KnS, T, C, D, Z(V)	DIV,DIVP: 7steps DDIV, DDIVP: 13 steps
		See page 4-46 for more details regarding floating point format.		Note: Z(V) may **NOT** be used for 32 bit operation	
		When using M8023 to subtract floating point data, only double word (32 bit) data registers (D) or constants (K/H) may be used.used to perform			

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation: (Applicable to all units)

The primary source (S_1) is divided by the secondary source (S_2). The result is stored in the destination (D). Note the normal rules of algebra apply.

Points to note:

a) When operating the DIV instruction in 16bit mode, two 16 bit data sources are divided into each other. They produce two 16 bit results. The device identified as the destination address is the lower of the two devices used to store the these results.
This storage device will actually contain a record of the number of whole times S_2 will divide into S_1 (the quotient).
The second, following destination register contains the remained left after the last whole division (the remainder). Using the previous example with some test data:
51 (D0) ÷ 10 (D2) = 5(D4) 1(D5)
This result is interpreted as 5 whole divisions with 1 left over ($5 \times 10 + 1 = 51$).

b) When operating the DIV instruction in 32 bit mode, two 32 bit data sources are divided into each other. They produce two 32 bit results. The device identified as the destination address is the lower of the two devices used to store the quotient and the following two devices are used to store the remainder, i.e. if D30 was selected as the destination of 32 bit division operation then D30, D31 would store the quotient and D32, D33 would store the remainder. If the location of the destination device is smaller than the obtained result, then only the portion of the result which directly maps to the destination area will be written. If bit devices are used as the destination area, no remainder value is calculated.

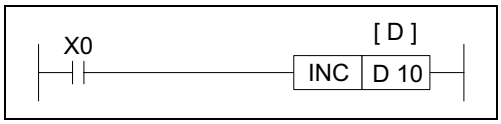
c) If the value of the source device S_2 is 0 (zero) then an operation error is executed and the operation of the DIV instruction is cancelled.

5.3.5 INC (FNC 24)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		D	
INC FNC 24 (Increment) ✈	The designated device is incremented by 1 on every execution of the instruction	KnY, KnM, KnS, T, C, D, V, Z Standard V,Z rules apply for 32 bit operation	INC,INCP: 3 steps DINC, DINCP: 5 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

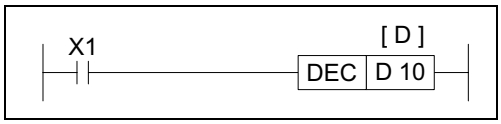
On every execution of the instruction the device specified as the destination D, has its current value incremented (increased) by a value of 1.
In 16 bit operation, when +32,767 is reached, the next increment will write a value of -32,768 to the destination device.
In 32 bit operation, when +2,147,483,647 is reached the next increment will write a value of -2,147,483,648 to the destination device.
In both cases there is no additional flag to identify this change in the counted value.

5.3.6 DEC (FNC 24)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		D	
DEC FNC 25 (Decrement) ✈	The designated device is decremented by 1 on every execution of the instruction	KnY, KnM, KnS, T, C, D, V, Z Standard V,Z rules apply for 32 bit operation	DEC,DECP: 3 steps DDEC, DDECP: 5 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

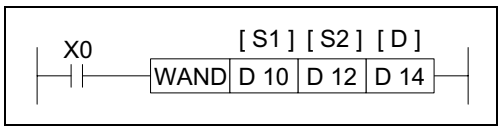
On every execution of the instruction the device specified as the destination D, has its current value decremented (decreased) by a value of 1.
In 16 bit operation, when -32,768 is reached the next increment will write a value of +32,767 to the destination device.
In 32 bit operation, when -2,147,483,648 is reached the next increment will write a value of +2,147,483,647 to the destination device.
In both cases there is no additional flag to identify this change in the counted value.

5.3.7 WAND (FNC 26)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
WAND FNC 26 (Logical word AND)	A logical AND is performed on the source devices - result stored at destination	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z	WAND,WANDP: 7 steps DAND, DANDP: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The bit patterns of the two source devices are analyzed (the contents of $S_2$is compared against the contents of S_1). The result of the logical AND analysis is stored in the destination device (D).

The following rules are used to determine the result of a logical AND operation. This takes place for every bit contained within the source devices:

General rule: (S_1) Bit n WAND (S_2) Bit n = (D) Bit n

1 WAND 1 = 1 0 WAND 1 = 0

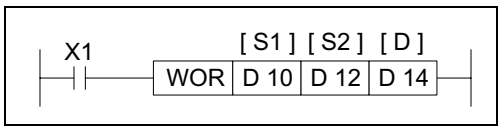
1 WAND 0 = 0 0 WAND 0 = 0

5.3.8 WOR (FNC 27)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
WOR FNC 27 (Logical word OR)	A logical OR is performed on the source devices - result stored at destination	K,H, KnX,KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z	WOR,WORP: 7 steps DOR, DORP: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The bit patterns of the two source devices are analyzed (the contents of $S_2$is compared against the contents of S_1). The result of the logical OR analysis is stored in the destination device (D).

The following rules are used to determine the result of a logical OR operation. This takes place for every bit contained within the source devices:

General rule: (S_1) Bit n WOR (S_2) Bit n = (D) Bit n

1 WOR 1 = 1 0 WOR 1 = 1

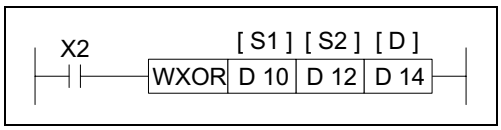
1 WOR 0 = 1 0 WOR 0 = 0

5.3.9 WXOR (FNC 28)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
WXOR FNC 28 (Logical exclusive OR)	A logical XOR is performed on the source devices - result stored at destination	K, H KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z	WXOR, WXORP: 7 steps DXOR,DXORP 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The bit patterns of the two source devices are analyzed (the contents of S_2 is compared against the contents of S_1). The result of the logical XOR analysis is stored in the destination device (D).

The following rules are used to determine the result of a logical XOR operation. This takes place for every bit contained within the source devices:

General rule: (S_1)Bit $_n$ WXOR (S_2)Bit $_n$ = (D)Bit $_n$

1 WXOR 1 = 0 0 WXOR 1 = 1

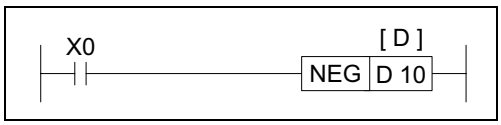
1 WXOR 0 = 1 0 WXOR 0 = 0

5.3.10 NEG (FNC 29)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		D	
NEG FNC 29 (Negation) ✈	Logically inverts the contents of the designated device	KnY, KnM, KnS, T, C, D, V, Z	NEG,NEGP: 3 steps DNEG, DNEGP: 5 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The bit pattern of the selected device is inverted. This means any occurrence of a ‘1’ becomes a ‘0’ and any occurrence of a ‘0’ will be written as a ‘1’. When this is complete, a further binary 1 is added to the bit pattern. The result is the total logical sign change of the selected devices contents, e.g. a positive number will become a negative number or a negative number will become a positive.

MEMO

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.4 Rotation And Shift - Functions 30 to 39

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

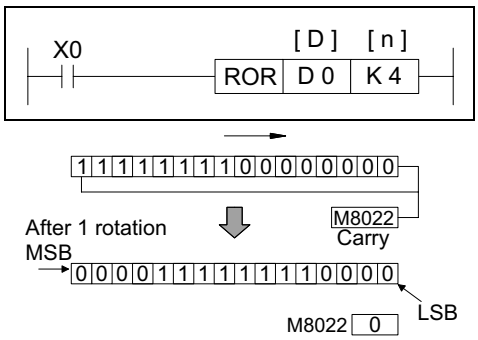
✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.4.1 ROR (FNC 30)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		D	n	
ROR FNC 30 (Rotation right) ✈	The bit pattern of the destination device is rotated 'n' places to the right on every execution	KnY, KnM, KnS, T, C, D, V, Z Note: 16 bit operation Kn=K4, 32 bit operation Kn=K8	K, H, ⊠ Note: 16 bit operation n≤ 16 32 bit operation n≤ 32	ROR, RORP: 5 steps DROR, DRORP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Carry M8022



Operation:

The bit pattern of the destination device (D) is rotated n bit places to the right on every operation of the instruction.

The status of the last bit rotated is copied to the carry flag M8022.

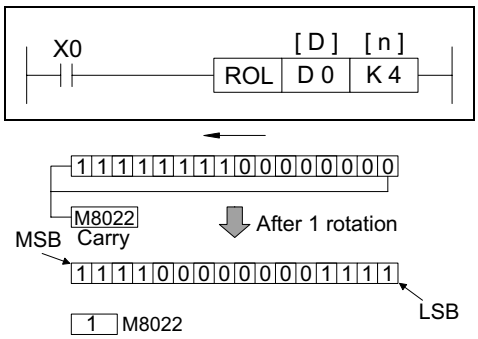
The example shown left is based on the instruction noted above it, where the bit pattern represents the contents of D0.

5.4.2 ROL (FNC 31)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
ROL FNC 31 (Rotation left) ✈	The bit pattern of the destination device is rotated 'n' places to the left on every execution	KnY, KnM, KnS, T, C, D, V, Z Note: 16 bit operation Kn= K4, 32 bit operation Kn= K8	K, H, ⊠ Note: 16 bit operation n ≤ 16 32 bit operation n≤ 32	ROL,ROLP: 5 steps DROL, DROLP: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Carry M8022



Operation:

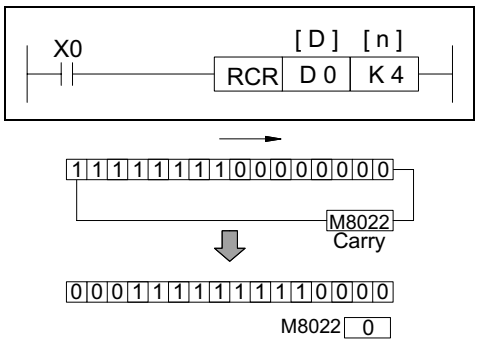
The bit pattern of the destination device (D) is rotated n bit places to the left on every operation of the instruction.
The status of the last bit rotated is copied to the carry flag M8022.
The example shown left is based on the instruction noted above it, where the bit pattern represents the contents of D0.

5.4.3 RCR (FNC 32)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		D	n	
RCR FNC 32 (Rotation right with carry) ✈	The contents of the destination device are rotated right with 1 bit extracted to the carry flag	KnY, KnM, KnS, T, C, D, V, Z Note: 16 bit operation Kn= K4, 32 bit operation Kn=K8	K, H, ⊠ Note: 16 bit operation n≤ 16 32 bit operation n≤ 32	RCR,RCRP: 5 steps DRCR, DRCRP: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Carry M8022



Operation:

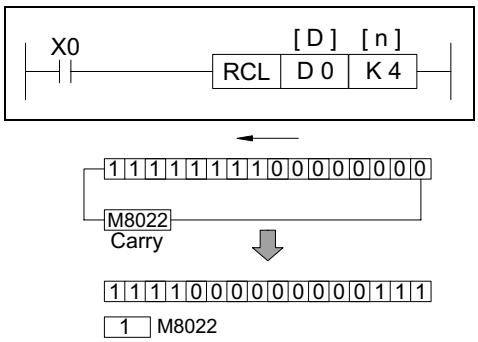
The bit pattern of the destination device (D)is rotated n bit places to the right on every operation of the instruction.
The status of the last bit rotated is moved into the carry flag M8022. On the following operation of the instruction M8022 is the first bit to be moved back into the destination device.
The example shown left is based on the instruction noted above it, where the bit pattern represents the contents of D0.

5.4.4 RCL (FNC 33)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
RCL FNC 33 (Rotation left with carry) ✈	The contents of the destination device are rotated left with 1 bit extracted to the carry flag	KnY, KnM, KnS, T, C, D, V, Z Note: 16 bit operation Kn= K4, 32 bit operation Kn= K8	K, H, ⊠ Note: 16 bit operation n≤ 16 32 bit operation n≤ 32	RCL, RCLP: 5 steps DRCL, DRCLP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Carry M8022



Operation:

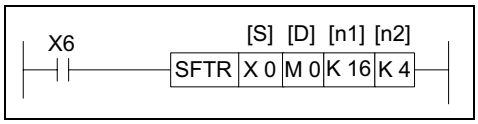
The bit pattern of the destination device (D)is rotated n bit places to the left on every operation of the instruction.
The status of the last bit rotated is moved into the carry flag M8022. On the following operation of the instruction M8022 is the first bit to be moved back into the destination device.
The example shown left is based on the instruction noted above it, where the bit pattern represents the contents of D0.

5.4.5 SFTR (FNC 34)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S	D	n1	n2	
SFTR FNC 34 (Bit shift right) ✈	The status of the source devices are copied to a controlled bit stack moving the existing data to the right	X, Y, M, S	Y, M, S	K,H, ⊠ Note: FX users: $n_2 \le n_1 \le 1024$ FX_0, FX_{0N} users: $n_2 \le n_1 \le 512$		SFTR,SFTRP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

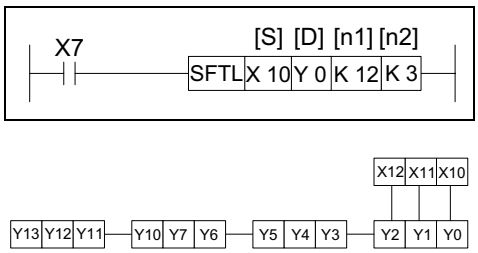
The instruction copies n_2 source devices to a bit stack of length n_1. For every new addition of n_2 bits, the existing data within the bit stack is shifted n_2 bits to the right. Any bit data moving to a position exceeding the n_1 limit is diverted to an overflow area. The bit shifting operation will occur every time the instruction is processed unless it is modified with either the pulse suffix or a controlled interlock.

5.4.6 SFTL (FNC 35)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S	D	n1	n2	
SFTL FNC 35 (Bit shift left) ✈	The status of the source devices are copied to a controlled bit stack moving the existing data to the left	X, Y, M, S	Y, M, S	K,H, ⊠ Note: FX users: $n_2 \le n_1 \le 1024$ FX_0,FX_{0N} users: $n_2 \le n_1 \le 512$		SFTL,SFTLP: 9steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

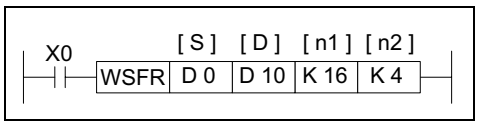
The instruction copies n_2 source devices to a bit stack of length n_1. For every new addition of n_2 bits, the existing data within the bit stack is shifted $n_2$bits to the left. Any bit data moving to a position exceeding the n_1 limit is diverted to an overflow area. The bit shifting operation will occur every time the instruction is processed unless it is modified with either the pulse suffix or a controlled interlock.

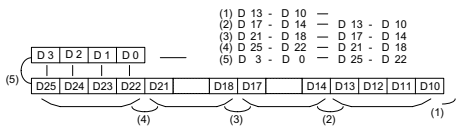
5.4.7 WSFR (FNC 36)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S	D	n1	n2	
WSFR FNC 36 (Word shift right) ✈	The value of the source devices are copied to a controlled word stack moving the existing data to the right	KnX, KnY, KnM,KnS, T, C, D	KnY,KnM, KnS, T, C, D	K,H, ⊠ Note: FX users: $n_2 \leq n_1 \leq 512$		WSFR, WSFRP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P





Operation:

The instruction copies n_2 source devices to a word stack of length n_1. For each addition of n_2 words, the existing data within the word stack is shifted $n_2$words to the right. Any word data moving to a position exceeding the n_1 limit is diverted to an overflow area. The word shifting operation will occur every time the instruction is processed unless it is modified with either the pulse suffix or a controlled interlock.

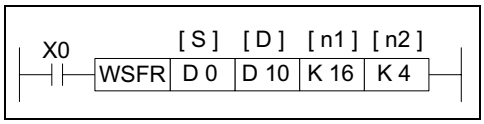
Note: when using bit devices as source (S) and destination (D) the Kn value must be equal.

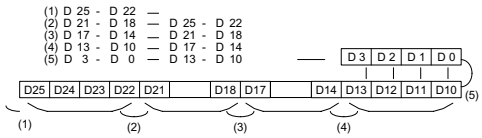
5.4.8 WSFL (FNC 37)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S	D	n1	n2	
WSFL FNC 37 (Word shift left) ✈	The value of the source devices are copied to a controlled word stack moving the existing data to the left	KnX, KnY, KnM,KnS, T, C, D	KnY,KnM, KnS, T, C, D	K,H, ⊠ Note: FX users: $n_2 \leq n_1 \leq 512$		WSFL, WSFLP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P





Operation:

The instruction copies n_2 source devices to a word stack of length n_1. For each addition of n_2 words, the existing data within the word stack is shifted $n_2$words to the left. Any word data moving to a position exceeding the n_1 limit is diverted to an overflow area. The word shifting operation will occur every time the instruction is processed unless it is modified with either the pulse suffix or a controlled interlock.

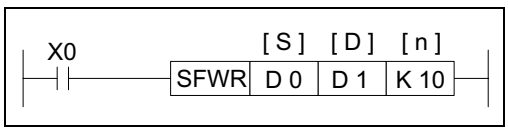
Note: when using bit devices as source (S) and destination (D) the Kn value must be equal.

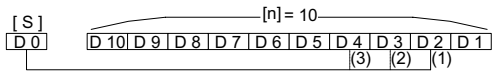
5.4.9 SFWR (FNC 38)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D	N	
SFWR FNC 38 (Shift register write) ✈	This instruction creates and builds a FIFO stack n devices long -must be used with SFRD FNC 39	K, H, KnX, KnY, KnM,KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D,	K, H, ⊠ Note: 2≤ n≤ 512	SFWR, SFWRP: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Carry M8022





Operation:

The contents of the source device (S) are written to the FIFO stack. The position of insertion into the stack is automatically calculated by the PLC.
The destination device (D) is the head address of the FIFO stack. The contents of D identify where the next record will be stored (as an offset from D_{+1}).

If the contents of D exceed the value “n-1” (n is the length of the FIFO stack) then insertion into

the FIFO stack is stopped. The carry flag M8022 is turned ON to identify this situation.

Points to note:

a) FIFO is an abbreviation for ‘First-In/ First-OUT’.

b) Although n devices are assigned for the FIFO stack, only n_{-1} pieces of information may be written to that stack. This is because the head address device (D) takes the first available register to store the information regarding the next data insertion point into the FIFO stack.

c) Before starting to use a FIFO stack ensure that the contents of the head address register (D) are equal to ‘0’ (zero).

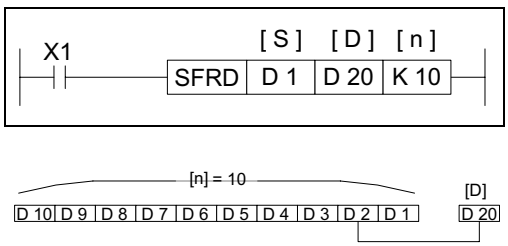
d) This instruction should be used in conjunction with SFRD FNC 39. The n parameter in both instructions should be equal.

5.4.10 SFRD (FNC 39)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D	n	
SFRD FNC 39 (Shift register read) ✈	This instruction reads and reduces FIFO stack- must be used with SFWR FNC 38	KnY, KnM, KnS, T, C, D, KnY, KnM,KnS, T, C, D	KnY, KnM, KnS, T, C, D, KnY, KnM,KnS, T, C, D, V, Z	K,H, ⊠ Note: 2≤ n≤ 512	SFRD, SFRDP: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Zero M8020



Operation:

The source device (S) identifies the head address of the FIFO stack. Its contents reflect the last entry point of data on to the FIFO stack, i.e. where the end of the FIFO is (current position).
This instruction reads the first piece of data from the FIFO stack (register S_{+1}), moves all of the data within the stack 'up' one position to fill the read area and decrements the contents of the FIFO head address (S) by 1. The read data is written to the destination device (D).
When the contents of the source device (S) are equal to '0' (zero), i.e. the FIFO stack is empty, the flag M8020 is turned ON.

Points to note:

a) FIFO is an abbreviation for 'First-In/ First-OUT'.

b) Only n_{-1} pieces of data may be read from a FIFO stack. This is because the stack requires that the first register, the head address (S) is used to contain information about the current length of the FIFO stack.

c) This instruction will always read the source data from the register S_{+1}.

d) This instruction should be used in conjunction with SFWR FNC 38. The n parameter in both instructions should be equal.

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.5 Data Operation - Functions 40 to 49

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

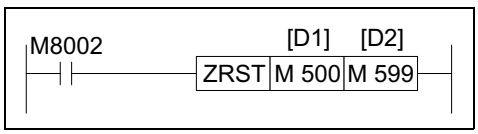
⊠ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.5.1 ZRST (FNC 40)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
ZRST FNC 40 (Zone Reset)	Used to reset a range of like devices in one operation	Y, M,S, T, C, D Note: D_1must be less than or equal (≤) to D_2. Standard and High speed counters cannot be mixed.		ZRST, ZRSTP: 5 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The range of devices, inclusive of those specified as the two destinations are reset, i.e. for data devices the current value is set to 0 (zero) and for bit elements, the devices are turned OFF, i.e. also set to 0 (zero).

The specified device range cannot contain mixed device types, i.e. C000 specified as the first destination device (D_1) cannot be paired with T199 as the second destination device (D_2). When resetting counters, standard and high speed counters cannot be reset as part of the same range.

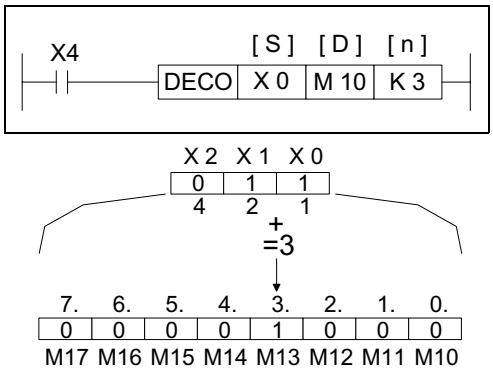
If D_1 is greater than (>) D_2 then only device D_1 is reset.

5.5.2 DECO (FNC 41)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D	n	
DECO FNC 41 (Decode)	Source data value Q identifies the Qth bit of the destination device which will be turned ON	K, H, X, Y, M,S, T, C, D, V, Z	Y, M, S, T, C, D	K, H, ✉ Note: D= Y,M,S then n range = 1-8 D= T,C,D then n range = 1-4 n= 0, then no processing	DECO, DECOP: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

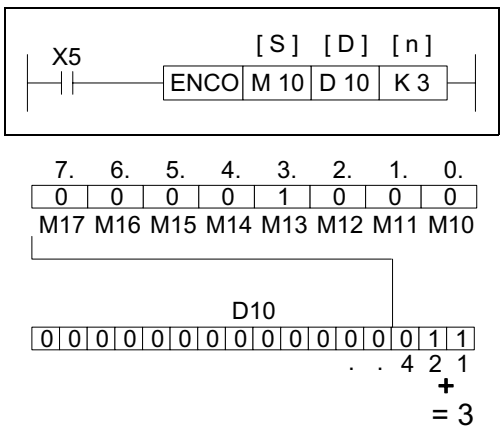
Source data is provided by a combination of operands S and n. Where S specifies the head address of the data and n, the number of consecutive bits. The source data is read as a single number (binary to decimal conversion) Q. The source number Q is the location of a bit within the destination device (D) which will be turned ON (see example opposite). When the destination device is a data device n must be within the range 1 to 4 as there are only 16 available destination bits in a single data word. All unused data bits within the word are set to 0.

5.5.3 ENCO (FNC 42)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D	n	
ENCO FNC 42 (Encode)	Then location of the highest active bit is stored as a numerical position from the head address	X, Y, M, S, T, C, D, V, Z	T, C, D, V, Z	K, H, ⊠ Note: S=X, Y, M, S then n range=1-8 S= T,C,D then n range = 1-4 n = 0, then no processing	ENCO, ENCOP: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The highest active bit within the readable range has its location noted as a numbered offset from the source head address (S). This is stored in the destination register (D).

Points to note:

a) The readable range is defined by the largest number storable in a binary format within the number of destination storage bits specified by n, i.e. if n was equal to 4 bits a maximum number within the range 0 to 15 can be written to the destination device. Hence, if bit devices were being used as the source data, 16 bit devices would be used, i.e. the head bit device and 15 further, consecutive devices.

b) If the stored destination number is 0 (zero) then the source head address bit is ON, i.e. the active bit has a 0 (zero) offset from the head address. However, if NO bits are ON within the source area, 0 (zero) is written to the destination device and an error is generated.

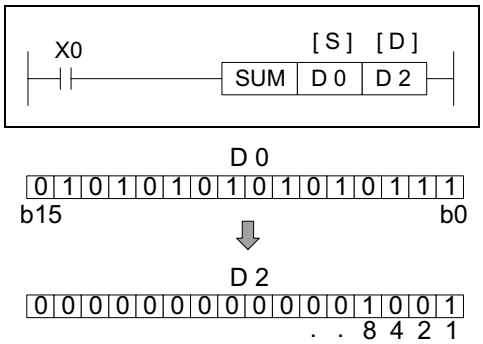
c) When the source device is a data or word device n must be taken from the range 1 to 4 as there are only 16 source bits available within a single data word.

5.5.4 SUM (FNC 43)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
SUM FNC 43 (Sum of active bits)	The number (quantity) of active bits in the source data is stored in the destination device	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	SUM,SUMP: 7 steps DSUM, DSUMP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Zero M8020



Operation:

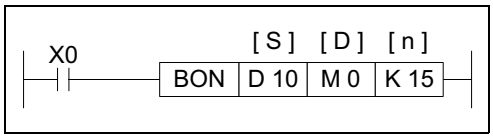
The number of active (ON) bits within the source device (S), i.e. bits which have a value of “1" are counted. The count is stored in the destination register (D). If a double word format is used, both the source and destination devices use 32 bit, double registers. The destination device will always have its upper 16 bits set to 0 (zero) as the counted value can never be more than 32.
If no bits are ON then zero flag, M8020 is set.

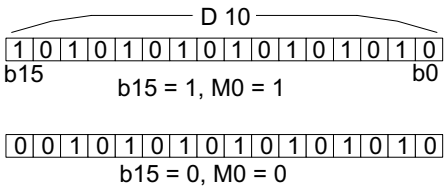
5.5.5 BON (FNC 44)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D	n	
BON FNC 44 (Check specified bit status)	The status of the specified bit in the source device is indicated at the destination	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	Y, M, S	K,H, ⊠ Note: 16 bit operation n=0 to 15 32 bit operation n=0 to 31	BON, BONP: 7steps DBONP, DBON: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P





Operation:

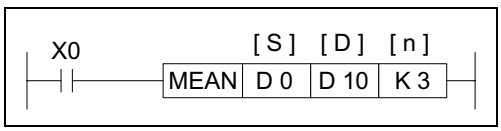
A single bit position (n) is specified from within a source device/area (S). n could be regarded as a specified offset from the source head address (S), i.e. 0 (zero) being the first device (a 0 offset) where as an offset of 15 would actually be the 16th device.
If the identified bit becomes active, i.e. ON, the destination device (D) is activated to “flag” the new status.
The destination device could be said to act as a mirror to the status of the selected bit source.

5.5.6 MEAN (FNC 45)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D	n	
MEAN FNC 45 (Mean)	Calculates the mean of a range of devices	KnX, KnY, KnM, KnS, T, C, D	KnY, KnM, KnS, T, C, D, V, Z	K,H, ⊠ Note: n= 1 to 64	MEAN, MEANP: 7 steps DMEAN, DMEANP: 13steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



General rule

$$D = \frac{\sum_{S_0}^{S_n} S}{n} = \frac{(S_0 + S_1 + < + S_n)}{n}$$

Example

$$D10 = \frac{(D0) + (D1) + (Dn)}{3}$$

Operation:

The range of source data is defined by operands S and n. S is the head address of the source data and n specifies the number of consecutive source devices used.
The value of all the devices within the source range is summed and then divided by the number of devices summed, i.e. n. This generates an integer mean value which is stored in the destination device (D). The remainder of the calculated mean is ignored.

Points to note:

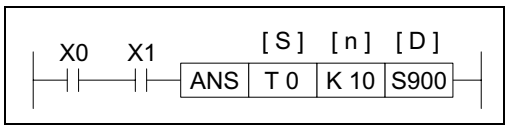
If the source area specified is actually smaller than the physically available area, then only the available devices are used. The actual value of n used to calculate the mean will reflect the used, available devices. However, the value for n which was entered into the instruction will still be displayed. This can cause confusion as the mean value calculated manually using this original n value will be different from that which is displayed.
If the value of nis specified outside of the stated range (1 to 64) an error is generated.

5.5.7 ANS (FNC 46)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D	n	
ANS FNC 46 (Timed annunciator set)	This instruction starts a timer. Once timed out the selected annunciator flag is set ON	T Note: available range T0 to T199	S Note: annunciator range S900 to S999	K ☒ Note: n range 1 to 32,767 - in units of 100msec	ANS: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

This instruction, when energized, starts a timer (S) for n,100 msec. When the timer completes its cycle the assigned annunciator (D) is set ON.

If the instruction is switched OFF during or after completion of the timing cycle the timer is automatically reset. However, the current status of the annunciator coil remains unchanged.

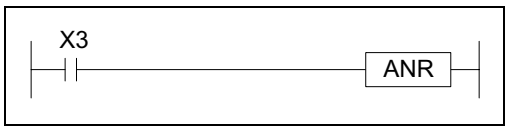
Note: This is only one method of driving annunciator coils, others such as direct setting can also be used.

5.5.8 ANR (FNC 47)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		D	
ANR FNC 47 (Annunciator reset) ✈	The lowest active annunciator is reset on every operation of this instruction	N/A	ANR,ANRP: 1step

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

Annunciators which have been activated are sequentially reset one-by-one, each time the ANR instruction is operated.

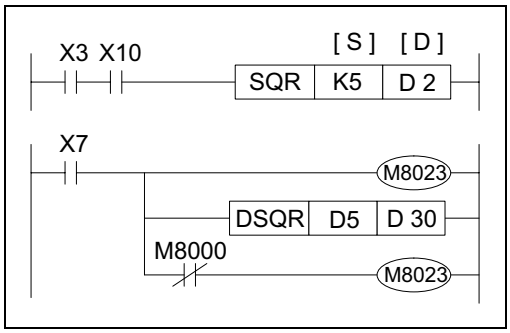
If the ANR instruction is driven continuously it will carry out its resetting operation on every program scan unless it is modified by the pulse, P prefix or by a user defined program interlock.

5.5.9 SQR (FNC 48)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
SQR FNC 48 (Square root)	Performs a mathematical square root e.g.: $D=\sqrt{S}$	K,H,D	D	SQR, SQRP: 5 steps DSQR, DSQRP: 9 steps
		When using M8023 in float mode, only double word (32bit) data can be processed. See page 4- 46 for more details regarding floating point.		

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Zero M8020 Borrow M8021



Operation 1:

This instruction performs a square root operation on source data (S) and stores the result at destination device (D). The operation is conducted entirely in whole integers rendering the square root answer rounded to the lowest whole number. For example, if (S) = 154, then (D) is calculated as being 12. M8020 is set ON when the square root operation result is equal to zero. Answers with rounded values will activate M8021.

Operation 2: This function is equivalent to FNC 127 ESQR This operation is similar to Operation 1. However, it is only activated when the mode setting float flag, M8023 is used. This then allows the SQR instruction to process answers in floating point format. The source data (S) must either be supplied in floating point format for data register use, or it can be supplied as a constant (K,H). When constants are used as a source, they are automatically converted to floating point format. Operation 2 is only valid for double word (32 bit) operation, hence both (S) and (D) will be 32 bit values and the SQR instruction will be entered as DSQR or DSQRP.

General note:

Performing any square root operation (even on a calculator) on a negative number will result in an error. This will be identified by special M coil M8067 being activated:

$\sqrt{-168}$ = Error and M8067 will be set ON

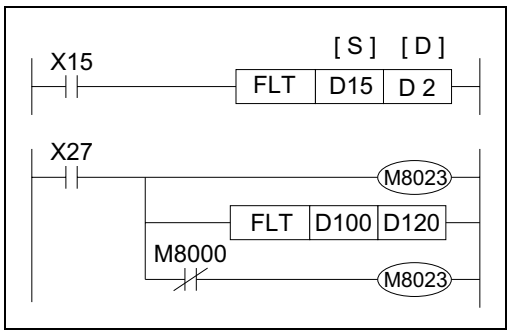
This is true for both operating modes.

5.5.10 FLT (FNC 49)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
FLT FNC 49 (Floating point)	Used to convert data to and from floating point format	D	D	FLT, FLTP: 5 steps DFLT, DFLTP: 9 steps
		M8023 = OFF data is converted from decimal to floating point format M8023 = ON data is converted from floating point format to decimal format		

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

When the float instruction is used without the float flag (M8023 = OFF) the source data (S) is converted in to an equivalent value stored in float format at the destination device (D).
Please note that two consecutive devices (D and D_{+1}) will be used to store the converted float number. This is true regardless of the size of the source data (S), i.e. whether (S) is a single device (16 bits) or a double device (32 bits) has no effect on the number of destination devices (D) used to store the floating point number. Examples:

Decimal source data (S)	Floating point destination value (D)
1	1
-26700	-2.67×10^4
404	4.04×10^2

MEMO

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.6 High Speed Processing - Functions 50 to 59

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

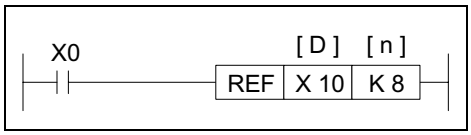
✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.6.1 REF (FNC 50)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		D	n	
REF FNC 50 (Refresh) ✈	Forces an immediate update of inputs or outputs as specified	X, Y ⊠ Note: D should always be a multiple of 10, i.e. 00, 10, 20, 30 etc.	K, H ⊠ Note: n should always be a multiple of 8, i.e. 8, 16, 24, 32 etc.	REF, REFP: 5 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

Standard PLC operation processes output and input status between the END instruction of one program scan and step 0 of the following program scan. If an immediate update of the I/O device status is required the REF instruction is used. The REF instruction can only be used to update or refresh blocks of 8 (n) consecutive devices. The head address of the refreshed devices should always have its last digit as a 0 (zero), i.e. in units of 10.

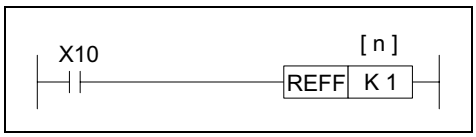
Note: A short delay will occur before the I/O device is physically updated, in the case of inputs a time equivalent to the filter setting, while outputs will delay for their set energized time.

5.6.2 REFF (FNC 51)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		n	
REFF FNC 51 (Refresh and filter adjust)	Inputs are refreshed, and their input filters are reset to the newly designated value	K, H, ⊠ Note: n= 0 to 60 msec (0 = 50µs) X000 to X007 (X000 to X017 for FX2N) are automatically designated when using this instruction	REFF, REFFP: 3 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

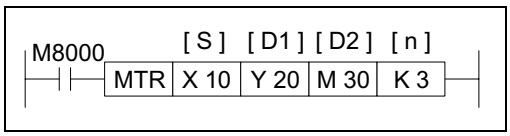
PLC's are provided with input filters to overcome problems generated by mechanical switch gear. However, as this involves ensuring a steady input signal is received for a fixed time duration, the use of input filters slows down the PLC response times. For high speed applications, especially where solid state switching provides the input signal, input filter times may be reduced. The default setting for the input filters is approximately 10 msec. Using this instruction input filter times of 0 to 60 msec may be selected. The setting '0' (zero) is actually 50 µsec. This is the minimum available setting. It is automatically selected when direct input, interrupts or high speed counting functions are used. The REFF instruction needs to be driven for each program scan if it is to be effective, otherwise, the standard 10 msec filter time is used.

5.6.3 MTR (FNC 52)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S	D1	D2	n	
MTR FNC 52 (Input matrix)	Multiplexes a bank of inputs into a number of sets of devices. Can only be used ONCE	X ☒	Y ☒	Y, M, S ☒	K, H, ☒ Note: n=2 to 8	MTR: 9 steps
		Note: These operands should always be a multiple of 10, i.e. 00, 10, 20, 30 etc.				

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Operation Complete M8029



Operation:

This instruction allows a selection of 8 consecutive input devices (head address S) to be used multiple (n) times, i.e. each physical input has more than one, separate and quite different (D_1) signal being processed. The result is stored in a matrix-table (head address D_2).

Points to note:

a) The MTR instruction involves high speed input/output switching. For this reason this instruction is only recommended for use with transistor output modules.

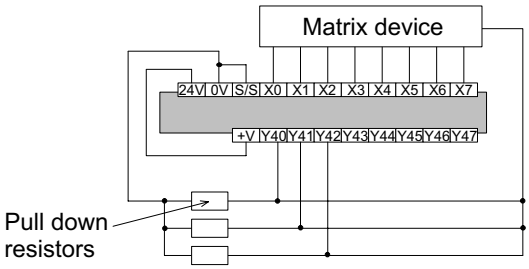
b) For the MTR instruction to operate correctly, it must be driven continuously. It is recommended that special auxiliary relay M8000, the PLC RUN status flag, is used. After the completion of the first full reading of the matrix, operation complete flag M8029 is turned ON. This flag is automatically reset when the MTR instruction is turned OFF.

c) Each set of 8 input signals are grouped into a 'bank' (there are n number of banks).

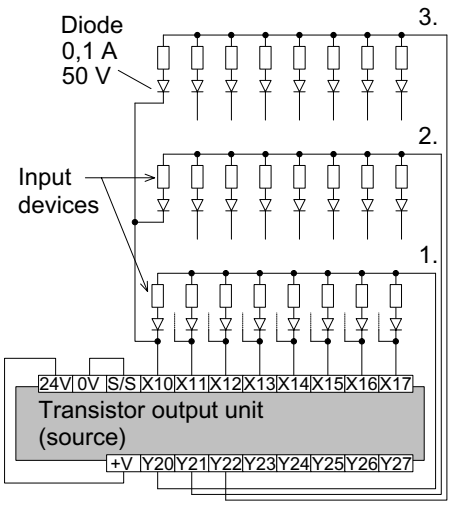
d) Each bank is triggered/selected by a dedicated output (head address D_1). This means the quantity of outputs from D_1, used to achieve the matrix are equal to the number of banks n. As there are now additional inputs entering the PLC these will each have a status which needs recording. This is stored in a matrix-table. The matrix-table starts at the head address D_2. The matrix construction mimics the same 8 signal by n bank configuration. Hence, when a certain input in a selected bank is read, its status is stored in an equivalent position within the result matrix-table.

e) The matrix instruction operates on an interrupt format, processing each bank of inputs every 20msec. This time is based on the selected input filters being set at 10msec. This would result in an 8 bank matrix, i.e. 64 inputs (8 inputs´ 8 banks) being read in 160msec.

If high speed inputs (ex. X0) is specified for operand S, the reading time of each bank becomes only 10msec, i.e. a halving of the reading speed. However, additional pull down resistors are required on the drive outputs to ensure the high speed reading does not detect any residual currents from the last operation. These should be placed in parallel to the input bank and should be of a value of approximately 3.3kΩ, 0.5W. For easier use, high speed inputs should not be specified at S.



f) Because this instruction uses a series of multiplexed signals it requires a certain amount of 'hard wiring' to operate. The example wiring diagram to the right depicts the circuit used if the previous example instruction was programmed. As a general precaution to aid successful operation diodes should be places after each input device (see diagram opposite). These should have a rating of 0.1A, 50V.



g) **Example Operation**
When output Y20 is ON only those inputs in the first bank are read. These results are then stored; in this example, auxiliary coils M30 to M37. The second step involves Y20 going OFF and Y21 coming ON. This time only inputs in the second bank are read. These results are stored in devices M40 to M47. The last step of this example has Y21 going OFF and Y22 coming ON. This then allows all of the inputs in the third bank to be read and stored in devices M50 to M57. The processing of this instruction example would take $20 \times 3 = 60$msec.

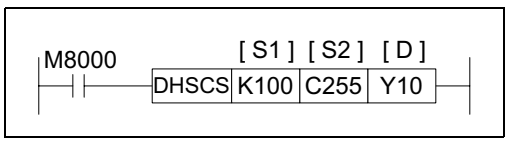
Notice how the resulting matrix-table does not use any of the ☆8 and ☆9 bit devices when state S or auxiliary M relays are used as the storage medium.

5.6.4 HSCS (FNC 53)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	n	
HSCS FNC 53 (High speed counter set)	Sets the selected output when the specified high speed counter value equals the test value	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	C Note: C = 235 to 254, or available high speed counters	Y, M, S Interrupt pointers I010 to I060 can be set.	DHSCS: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The HSCS set, compares the current value of the selected high speed counter (S_2)against a selected value (S_1). When the counters current value changes to a value equal to $S_1$the device specified as the destination (D)is set ON. The example above shows that Y10 would be set ON only when C255's value stepped from 99-100 OR 101-100. If the counters current value was forced to equal 100, output Y10 would **NOT** be set ON.

Points to note:

a) It is recommended that the drive input used for the high speed counter functions; HSCS, HSCR, HSCZ is the special auxiliary RUN contact M8000.

b) If more than one high speed counter function is used for a single counter the selected flag devices (D) should be kept within 1 group of 8 devices, i.e. Y0-7, M10-17.

c) All high speed counter functions use an interrupt process, hence, all destination devices (D) are updated immediately.

Note:

For all units Max. 6 simultaneously active HSCS/R and HSZ instructions. Please remember that the use of high speed counter functions has a direct impact on the maximum allowable counting speed! See page 4-22 for further details.

Use of interrupt pointers

FX1S	FX1N	FX2N	FX2NC

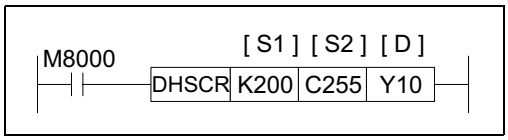
FX2N and FX2NC MPUs can use interrupt pointers I010 through I060 (6 points) as destination devices (D). This enables interrupt routines to be triggered directly when the value of the specified high speed counter reaches the value in the HSCS instruction.

5.6.5 HSCR (FNC 54)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
HSCR FNC 54 (High speed counter reset)	Resets the selected output when the specified high speed counter equals the test value	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	C Note: C = C235 to C255, or available high speed counters	Y, M, S C Note: If C, use same counter as S2	DHSCR: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

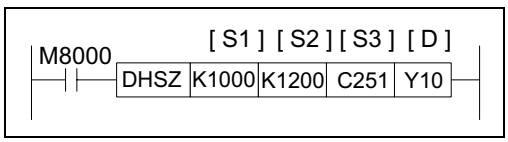
The HSCR, compares the current value of the selected high speed counter (S2) against a selected value (S1). When the counters current value changes to a value equal to S1, the device specified as the destination (D) is reset. In the example above, Y10 would be reset only when C255's value stepped from 199 to 200 or from 201 to 200. If the current value of C255 was forced to equal 200 by test techniques, output Y10 would **NOT** reset.
For further, general points, about using high speed counter functions, please see the subsection 'Points to note' under the HSCS (FNC 53). Relevant points are; a, b, and c.
Please also reference the note about the number of high speed instructions allowable.

5.6.6 HSZ (FNC 55)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S1	S2	S3	D	
HSZ FNC 55 (High speed zone compare)	Operation 1: The current value of a high speed counter is checked against a specified range	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		C Note: C = 235 to 255,	Y, M, S Note: 3 consecutive devices are used	DHSZ: 17 steps
	Operation 2: The designated range is held in a data table driving 'Y' outputs directly	D	K,H Using values from 1 to 128 (decimal)		M8130 (only) This flag can only be used with one DHSZ instr' at a time	
	Operation 3: The designated range is held in a data table driving PLSY frequencies directly using D8132				M8132 (only) This flag can only be used with one	

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation 1 - Standard: (Applicable to all units)
This instruction works in exactly the same way as the standard ZCP (FNC11). The only difference is that the device being compared is a high speed counter (specified as S3).
Also, all of the outputs (D) are updated immediately due to the interrupt operation of the DHSZ. It should be remembered that when a device is specified in operand D it is in fact a head address for 3 consecutive devices. Each one is used to represent the status of the current comparison, i.e. using the above example as a basis,

Y10 (D) C251 is less than S1, K1000 (S3< S1)
Y11 (D+1) C251 is greater than S1, K1000 but less than S2, K1200 (S3> S1, S3< S2)
Y12 (D+2) C251 is greater than S2, K1200 (S3> S2)

For further, general points, about using high speed counter functions please see the subsection 'Points to note' under the HSCS (FNC 52). Relevant points are; a, b, and c. Please also reference the note about the number of high speed instructions allowable.

Operation 2 - Using HSZ With A Data Table: (Applicable units: FX2N and FX2NC)
Operation 2 is selected when the destination device (D) is assigned special M coil M8130. This then allows devices (S1, S2) to be used to define a data table using (S1) as the head address and (S2) as the number of records in the table - maximum number of records is 128. Each record occupies 4 consecutive data registers proportioned in the following manner (for a single record of data registers D through D+3):

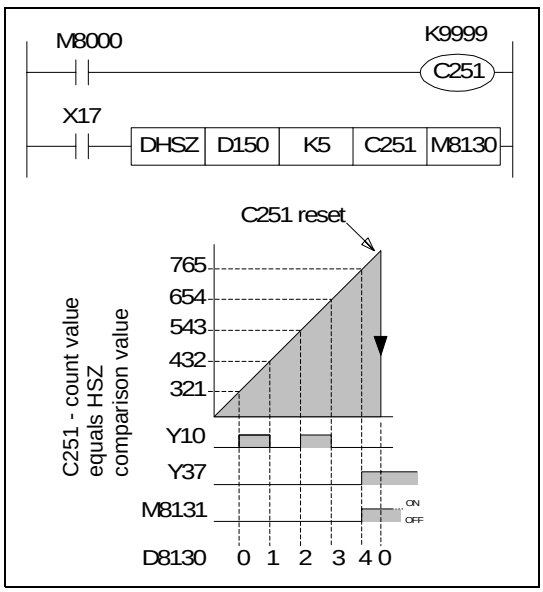
	Single Record	
Data registers	D, D+1	Used as a double (32 bit) data register to contain the comparison data
	D+2	Stores the I/O device number, in HEX, of the 'Y' Output device to be controlled, i.e. H10=Y10. Note: Hex digits A through F are not used.
	D+3	Stores the action (SET/RESET) to be performed on the Output device D+2. Note: For a SET (ON) operation D+3 must equal 1, for a RESET (OFF) D+3 must equal 0.

The following points should be read while studying the example on the right of the page. Please note, all normal rules associated with high speed counters still apply.

The data table is processed one 'record number' at a time, i.e only 1 record is ever active as the comparison data. The currently active record number is stored in data register D8130. As the comparison value for the active record is 'reached', the assigned 'Y' device is SET or RESET and the active 'Record number' is incremented by 1. Once all records in a data table have been processed, the current record pointer (D8130) is reset to 0 (the table is then ready to process again) and the operation complete flag M8131 is set ON.

Record number [D8130]	**Comparison value (lower/upper register)** [D, D+1]	**Selected 'Y' Output Device** [D+2]	**SET/RESET 'Y'Device** (1=SET, 0=RESET) [D+3]
0	[D150, D151] K321	[D152] H10 (Y10)	[D153] K1
1	[D154, D155] K432	[D156] H10 (Y10)	[D157] K0
2	[D158, D159] K543	[D160] H10 (Y10)	[D161] K1
3	[D162, D163] K765	[D164] H10 (Y10)	[D165] K0
4	[D166, D167] K765	[D168] H37 (Y37)	[D169] K1

If the high speed counter is reset (by program or hardware input), when it resumes counting and reaches the first record's comparison value, the M8131 flag will be reset. Both the status of M8131 and contents of D8130 are not editable by the user. If the DHSZ instruction is turned OFF then all associated flags are reset.
Care should be exercised when resetting the high speed counter or turning OFF the DHSZ instruct as all associated 'Y' output devices will remain in their last state, i.e. if an output was ON it will remain ON until independently reset by the user.
The data within inactive records can be changed during operation allowing data tables to be updated. Any change made is processed at the end of the current program scan. The HSZ instruction will continue to process only the active data record, i.e. it will not reset due to the updating of an inactive data record.



When the DHSZ instruction is initially activated it will not process a comparison until the following program scan as the CPU requires a slight time delay to initialize the comparison table.

Operation 3 - Combined HSZ and PLSY Operation: (Applicable units: FX_{2N} and FX_{2NC})
Operation 3 allows the HSZ and PLSY instructions to be used together as a control loop. This operation is selected when the destination device (D) is assigned special M coil M8132. This then allows devices (S_1, S_2) to be used to define a data table using (S_1) as the head address and (S_2) as the number of records in the table - maximum number of records is 128. Each record occupies 4 consecutive data registers (D through D_{+3}) proportioned in to two 32 bit data areas.

The first pair of data registers (D,D_{+1}) contain the comparison value for use with the high speed counter. The second pair of data registers (D_{+2},D_{+3}) contain a value (from 0 to 1000) which represents an output frequency in Hz. This value is loaded in to special data register D8132 when the comparison made by the DHSZ instruction gives a 'TRUE' output.

Record number [D8131]	Comparison value (lower/upper register) [D, D_{+1}]	Output Frequency For PLSY Instruction [D_{+2}, D_{+3}]
0	[D180, D181] K40	[D182, D183] K100
1	[D184, D185] K100	[D186, D187] K600
2	[D188, D189] K400	[D190, D191] K550
3	[D192, D193] K800	[D194, D195] K40
4	[D196, D197] K0	[D198, D199] K0

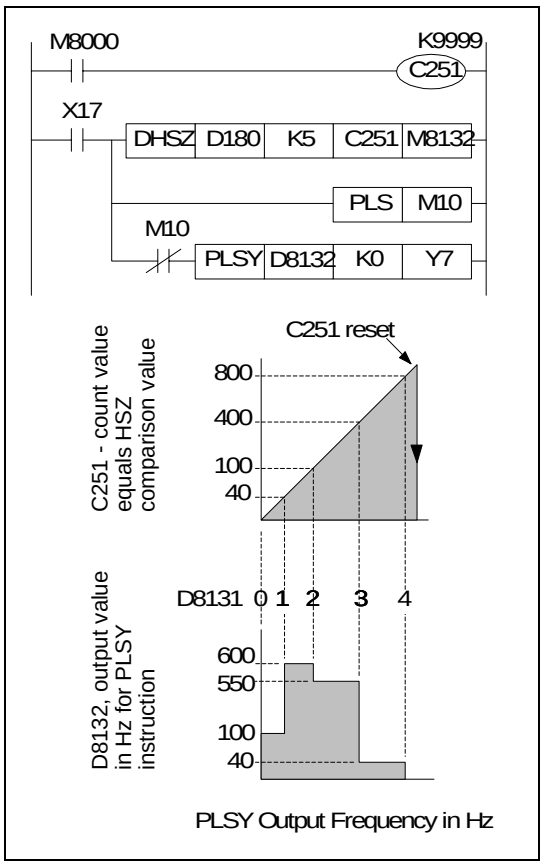
Special data register D8132 can be used as the source data for a PLSY (FNC57) output enabling the output to be varied with relative count data.

As with Operation 2 only one record in the data table is active at any one time. The current 'Record number' being processed is stored in data register D8131. To observe the current comparative value, data registers D8134 and D8135 should be monitored as a double word (32 bit) device.

Once the final entry in the data table has been processed, the operation complete flag M8133 is set ON and the record counter (D8131) cycles back to the first record.

It is recommended that if the high speed counter and PLSY operations form a closed loop that the last record entry in the data table is set to K0 for the comparison value and K0 for the PLSY output frequency. This will bring the controlled system to a stop and the 'Record number' counter will not be able to cycle back to the start of the data table until the associated high speed counter is reset by either pro-gram or hardware methods. This situation can be easily monitored by checking the paired data registers D8134 and D8135 for the '0' value.

It is recommended that the operation of the PLSY instruction is delayed for 1 scan to allow the DHSZ data table to be constructed on initial operation. A suggested program using a pulsed flag is shown in the example on this page.

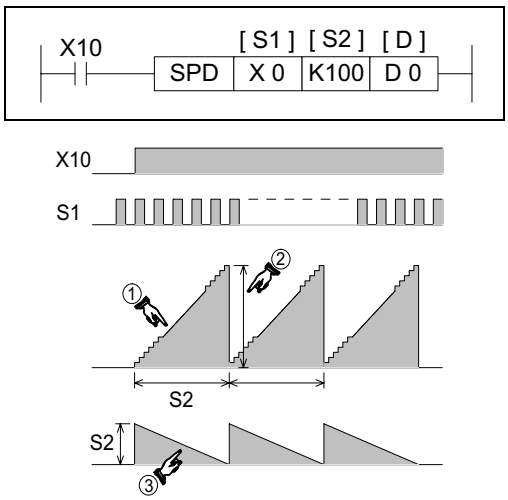


5.6.7 SPD (FNC 56)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
SPD FNC 56 (Speed detection)	Detects the number of 'encoder' pulses in a given time frame. Results can be used to calculate speed	X0 to X5	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	T, C, D, Z (V) Note: 3 consecutive devices are used. In the case of D= Z monitor D8028, D8029 and D8030	SPD: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The number of pulses received at S_1 are counted and stored in D_{+1}; this is the current count value. The counting takes place over a set time frame specified by S_2 in msec. The time remaining on the current 'timed count', is displayed in device D_{+2}. The number of counted pulses (of S_1) from the last timed count are stored in D. The timing chart opposite shows the SPD operation in a graphical sense. Note:

①: Current count value, device D_{+1}

②: Accumulated/ last count value, device D

③: Current time remaining in msec, device D+2

Points to note:

a) When the timed count frame is completed the data stored in D_{+1} is immediately written to D. D_{+1} is then reset and a new time frame is started.

b) Because this is both a high speed and an interrupt process only inputs X0 to X5 may be used as the source device S_1. However, the specified device for S_1 must **NOT** coincide with any other high speed function which is operating, i.e. a high speed counter using the same input. The SPD instruction is considered to act as a single phase counter.

c) Multiple SPD instructions may be used, but the identified source devices S_1 restrict this to a maximum of 6 times.

d) Once values for timed counts have been collected, appropriate speeds can be calculated using simple mathematics. These speeds could be radial speeds in rpm, linear speeds in M/ min it is entirely down to the mathematical manipulation placed on the SPD results. The following interpretations could be used;

$$\text{Linear speed N (km/h)} = \frac{3600 \times (D)}{n \times S_2} \times 10^3$$

where n = the number of linear encoder divisions per kilometer.

$$\text{Radial speed N (rpm)} = \frac{60 \times (D)}{n \times S_2} \times 10^3$$

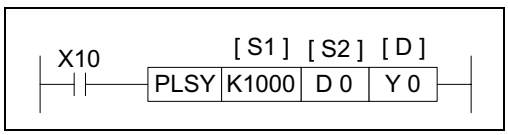
where n = the number of encoder pulses per revolution of the encoder disk.

5.6.8 PLSY (FNC 57)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
PLSY FNC 57 (Pulse Y output)	Outputs a specified number of pulses at a set frequency	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		Y Note: Y000 or Y001 only ⊠.	PLSY: 7 steps DPLSY: 13steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Operation Complete M8029



Operation:

A specified quantity of pulses S2 is output through device D at a specified frequency S1. This instruction is used in situations where the quantity of outputs is of primary concern.

Points to note:

a) FX1S/FX1N users may use frequencies of 1 to 132,767Hz (16-bit operation) and 1 to 100kHz (32-bit operation). FX2N/FX2NC users may use frequencies of 2 to 20kHz.

b) The maximum number of pulses: 16 bit operation: 1 to 32,767 pulses, 32 bit operation: 1 to 2,147,483,647 pulses.
Note: special auxiliary coil M8029 is turned ON when the specified number of pulses has been completed. The pulse count and completion flag (M8029) are reset when the PLSY instruction is de-energized. If “0" (zero) is specified the PLSY instruction will continue generating pulses for as long as the instruction is energized.

c) A single pulse is described as having a 50% duty cycle. This means it is ON for 50% of the pulse and consequently OFF for the remaining 50% of the pulse. The actual output is controlled by interrupt handling, i.e. the output cycle is NOT affected by the scan time of the program.

d) The data in operands S1 and S2 may be changed during execution. However, the new data in S2 will not become effective until the current operation has been completed, i.e. the instruction has been reset by removal of the drive contact.

e) Two FNC 57 (PLSY) can be used at the same time in a program to output pulses to Y000 and Y001 respectively. Or, only one FNC 57 PLSY and one FNC 59 PLSR can be used together in the active program at once, again outputting independent pulses to Y000 and Y001.

It is possible to use subroutines or other such programming techniques to isolate different instances of this instructions. In this case, the current instruction must be deactivated before changing to the new instance.

f) Because of the nature of the high speed output, transistor output units should be used with this instruction. Relay outputs will suffer from a greatly reduced life and will cause false outputs to occur due to the mechanical ‘bounce’ of the contacts.
To ensure a ‘clean’ output signal when using transistor units, the load current should be 200mA or higher with the FX2N Series. The load current should be 10 - 100mA with the FX1S/1N Series. It may be found that ‘pull up’ resistors will be required.

g) FX2N and FX2NC units can use the HSZ (FNC 55) instruction with the PLSY instruction when source device S1 is set to D8132. Please see page 5-59 for more details.

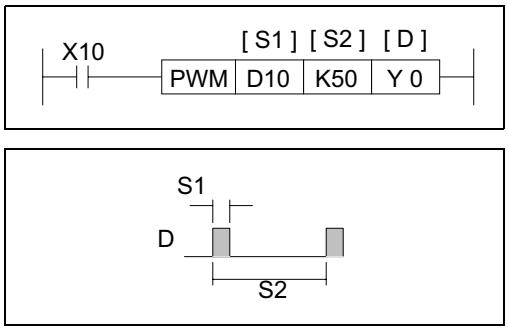
h) FX2N and FX2NC units can monitor the number of pulses output to Y0 using devices D8140 and D8141, and the number of output pulses output to Y1 using devices D8142 and D8143. The total number of pulses output can be monitored using D8136 and D8137.

5.6.9 PWM (FNC 58)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
PWM FNC 58 (Pulse width modulation)	Generates a pulse train with defined pulse characteristics	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z Note: S1 S2		Y Note: All units: Y000 or Y001 only ✉	PWM: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

A continuous pulse train is output through device D when this instruction is driven. The characteristics of the pulse are defined as:
The distance, in time (msec), between two identical parts of consecutive pulses (S_2).
And how long, also in time (msec), a single pulse will be active for (S_1).

Points to note:

a) Because this is a 16 bit instruction, the available time ranges for S_1 and S_2 are 1 to 32,767.

b) A calculation of the duty cycle is easily made by dividing S_1 by S_2. Hence S_1 cannot have a value greater than S_2 as this would mean the pulse is on for longer than the distance between two pulses, i.e. a second pulse would start before the first had finished. If this is programmed an error will occur.
This instruction is used where the length of the pulse is the primary concern.

c) The PWM instruction may only be used once in a users program.

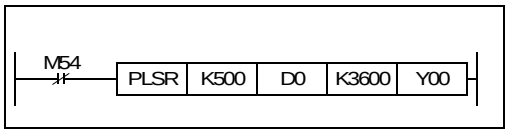
d) Because of the nature of the high speed output, transistor output units should be used with this instruction. Relay outputs will suffer from a greatly reduced life and will cause false outputs to occur due to the mechanical 'bounce' of the contacts. To ensure a 'clean' output signal when using transistor units, the load current should be 200mA or higher with the FX2N Series. The load current should be 10 - 100mA with the FX1S/1N Series. It may be found that 'pull up' resistors will be required.

5.6.10 PLSR (FNC 59)

FX1S	FX1N	FX2N	FX2NC

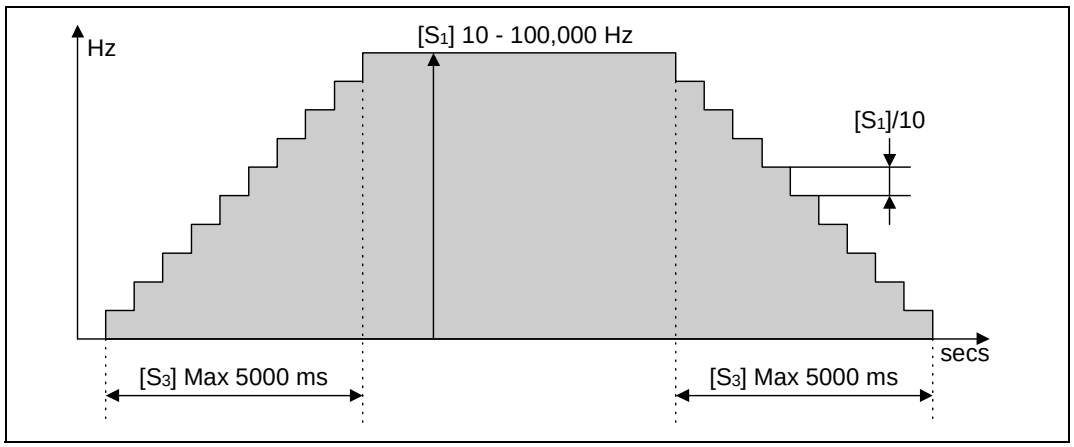
16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Operation Complete M8029

Mnemonic	Function	Operands				Program steps
		S1	S2	S3	D	
PLSR FNC 59 (Pulse ramp)	Outputs a specified number of pulses, ramping up to a set frequency and back down to stop	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z Note: S1 S2			Y FX2N users: Y000 or Y001 only.	PLSR: 9 steps DPLSR: 17 steps



Operation:

A specified quantity of pulses S_2 is output through device D. The output frequency is first ramped up in 10 steps to the maximum frequency S_1 in acceleration time S_3 ms, then ramped down to stop also in S_3 ms. This instruction is used to generate simple acc/dec curves where the quantity of outputs is of primary concern.



Points to Note:

a) FX2N/FX2NC users may use frequencies of 10 to 20,000Hz. FX1S/FX1N users may use frequencies of 10 to 100,000Hz. The frequency should be set to a multiple of 10. If not it will be rounded up to the next multiple of 10.
The acceleration and deceleration steps are set to 1/10 of the maximum frequency. Take this in to consideration to prevent slipping, when using stepping motors.

b) FX2N and FX2NC units with CPU of less than V3.00 and all FX1S, FX1N units,
maximum number of pulses: 16 bit operation: 110 to 32,767 pulses,
32 bit operation: 110 to 2,147,483,647 pulses.
Correct pulse output can not be guaranteed for a setting of 110 or less.
FX2N and FX2NC units with CPU of V3.00 or greater,
maximum number of pulses: 16 bit operation: 0 to 32,767 pulses,
32 bit operation: 0 to 2,147,483,647 pulses.
A setting of 110 pulses or less, or a frequency of $[S_1]/10$ will result in no acceleration.

c) The acceleration time must conform to the limitations described below.

d) The output device is limited to Y000 or Y001 only and should be transistor type.

e) Two FNC 59 (PLSR) can be used at the same time in a program to output pulses to Y000 and Y001 respectively. Or, only one FNC 57 PLSY and one FNC 59 PLSR can be used together in the active program at once, again outputting independent pulses to Y000 and Y001.

It is possible to use subroutines or other such programming techniques to isolate different instances of this instructions. In this case, the current instruction must be deactivated before changing to the new instance.

f) If the number of pulses is not enough to reach the maximum frequency then the frequency is automatically cut

g) Special auxiliary coil M8029 turns ON when the specified number of pulses has been completed. The pulse count and completion flag (M8029) are reset when the PLSR instruction is de-energized.

Acceleration time limitations

The acceleration time S_3 has a maximum limit of 5000 ms. However, the actual limits of S_3 are determined by other parameters of the system according to the following 4 points.

1) Set S3 to be more than 10 times the maximum program scan time (D8012).
If set to less than this, then the timing of the acceleration steps becomes uneven.

$$S_3 \geq \frac{90000}{S_1} \times 5$$

2) The following formula gives the minimum value for S3.

3) The following formula gives the maximum value for S3.

$$S_3 \leq \frac{S_2}{S_1} \times 818$$

4) The pulse output always increments in 10 step up to the maximum frequency as shown on the previous page.

If the parameters do not meet the above conditions, reduce the size of S_1.

- Possible output frequency is limited to 2 to 20,000 Hz for FX_{2N}/FX_{2NC}, and 10 to 100,000Hz for FX_{1S}/FX_{1N}. If either the maximum frequency or the acceleration step size are outside this limit then they are automatically adjusted to bring the value back to the limit.

- If the drive signal is switch off, all output stops. When driven ON again, the process starts from the beginning.

- Even if the operands are changed during operation, the output profile does not change. The new values take effect from the next operation.

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.7 Handy Instructions - Functions 60 to 69

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

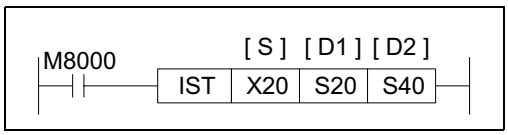
✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.7.1 IST (FNC 60)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	S3	
IST FNC 60 (Initial state)	Automatically sets up a multi-mode STL operating system	X, Y, M, S, Note: uses 8 consecutive devices	S, Note: FX0 users S20 to S63 FX0N users S20 to S127 FX users S20 to S899 D1must be lower than D2		IST: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

This instruction automatically sets up a multi-mode STL operating system. This consists of variations of 'manual' and 'automatic' operation modes.

Points to note:

a) The IST instruction automatically assigns and uses many bit flags and word devices; these are listed in the boxed column on the right of this page.

b) The IST instruction may only be used **ONCE**.
It should be programmed close to the beginning of the program, before the controlled STL circuits.

c) The required operation mode is selected by driving the devices associated with operands
S+0 through to S+4 (5 inputs). None of the devices within this range should be ON at the same time. It is recommended that these 'inputs' are selected through use of a rotary switch.
If the currently selected operating mode is changed before the 'zero return complete' flag (M8043) is set, all outputs will be turned OFF.

d) The 'zero position' is a term used to identify a datum position from where the controlled device, starts from and returns too after it has completed its task. Hence, the operating mode 'zero return', causes the controlled system to return to this datum.

Assigned devices

Indirect user selected devices:
S_{+0} Manual operation
S_{+1} Zero return
S_{+2} Step operation
S_{+3} One cycle operation
S_{+4} Cyclic operation
S_{+5} Zero return start
S_{+6} Automatic operation start
S_{+7} Stop

Initial states:
S0 initiates 'manual' operation
S1 initiates 'zero return' operation
S2 initiates 'automatic' operation

General states:
S10 to S19 'zero return' sequence
D_1 to D_2 'automatic return' sequence

Special bit flags:
M8040 = ON STL state transfer is inhibited
M8041 = ON initial states are enabled
M8042 = Start pulse given by start input
M8043 = ON zero return completed
M8044 = ON machine zero detected
M8047 = ON STL monitor enabled

The 'zero' position is sometimes also referred to as a home position, safe position, neutral position or a datum position.

e) The available operating modes are split into two main groups, manual and automatic. There are sub-modes to these groups. Their operation is defined as:

Manual

Manual (selected by device S+0)- Power supply to individual loads is turned ON and OFF by using a separately provided means, often additional push buttons.
Zero Return (selected by device S+1) -Actuators are returned to their initial positions when the Zero input (S+5) is given.

Automatic

One Step (selected by device S+2)- The controlled sequence operates automatically but will only proceed to each new step when the start input (S+6) is given.
One Cycle (selected by device S+3) - The controlled actuators are operated for **one** operation cycle. After the cycle has been completed, the actuators stop at their 'zero' positions. The cycle is started after a 'start' input (S+6) has been given.
A cycle which is currently being processed can be stopped at any time by activating the 'stop' input (S+7). To restart the sequence from the currently 'paused' position the start input must be given once more.
Automatic (selected by device S+4)-Fully automatic operation is possible in this mode. The programmed cycle is executed repeatedly when the 'start' input (S+6) is given. The currently operating cycle will not stop immediately when the 'stop' input (S+7)is given.
The current operation will proceed to then end of the current cycle and then stop its operation.

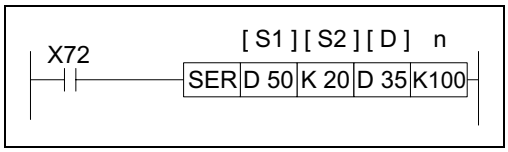
Note: Start, stop and zero inputs are often given by additional, manually operated push buttons.
Please note that the 'stop' input is only a program stop signal. It **cannot** be used as a replacement for an 'Emergency stop' push button. All safety, 'Emergency stop' devices should be hardwired systems which will effectively isolate the machine from operation and external power supplies. Please refer to local and national standards for applicable safety practices.

5.7.2 SER (FNC 61)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S1	S2	D	n	
SER FNC 61 (Search a Data Stack)	Generates a list of statistics about a single data value located/found in a data stack	KnX, KnY, KnM, KnS, T, C, D	KnX, KnY, KnM, KnS, T, C, D V, Z K, H	KnY, KnM, KnS T, C, D Note: 5 consecutive devices are used	K,H, D ⊠ Note: n= 1~256 for 16 bit operation n= 1~128 for 32 bit operation	SER, SERP: 9 steps DSER, DSERP: 17 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The SER instruction searches a defined data stack from head address S1, with a stack length n. The data searched for is specified in parameter S2 and the results of the search are stored at destination device D for 5 consecutive devices.

Destination device	Device description
D	Total number of occurrences of the searched value S2 (0 if no occurrences are found)
D+1	The position (within the searched data stack) of the first occurrence of the searched value S2
D+2	The position (within the searched data stack) of the last occurrence of the searched value S2
D+3	The position (within the searched data stack) of the smallest value found in the data stack (last occurrence is returned if there are multiple occurrences of the same value)
D+4	The position (within the searched data stack) of the largest value found in the data stack (last occurrence is returned if there are multiple occurrences of the same value)

Points to note:

a) Normal rules of algebra are used to determine the largest and smallest values, i.e. -30 is smaller than 6 etc.

b) If no occurrence of the searched data can be found then destination devices D, D+1 and D+2 will equal 0 (zero).

c) When using data register s as the destination device D please remember that 16 bit operation will occupy 5 consecutive, data registers but 32 bit operation will occupy 10 data registers in pairs forming 5 double words.

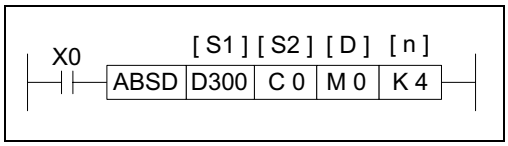
d) When multiple bit devices are used to store the result (regardless of 16 or 32 bit operation), only the specified size of group is written to for 5 consecutive occurrences, i.e. K1Y0 would occupy 20 bit devices from Y0 (K1 = 4 bit devices and there will be 5 groups for the 5 results). As the maximum data stack is 256 (0 to 255) entries long, the optimum group of bit devices required is K2, i.e. 8 bit devices.

5.7.3 ABSD (FNC 62)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S1	S2	D	n	
ABSD FNC 62 (Absolute drum sequencer)	Generates multiple output patterns in response to counter data	KnX, KnY, KnM, KnS, (in groups of 8) T, C, D	C	Y,M,S Note: n consecutive devices are used	K,H ⊠ Note: n≤ 64	ABSD: 9 steps DABSD: 17 steps. see f).
		Note: High speed counters are not allowed				

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

This instruction generates a variety of output patterns (there are n number of addressed outputs) in response to the current value of a selected counter, S2.

Points to note:

a) The current value of the selected counter (S2) is compared against a user defined data table. This data table has a head address identified by operand S1. S1should always have an even device number.

b) For each destination bit (D) there are two consecutive values stored in the data table. The first allocated value represents the event number when the destination device (D) will be turned ON. The second identifies the reset event. The data table values are allocated as a consecutive pair for each sequential element between D and D+n.

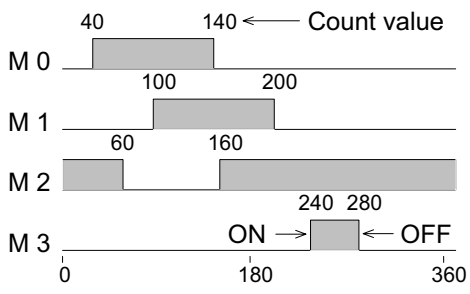
c) The data table has a length equal to 2 × n data entries. Depending on the format of the data table, a single entry can be one data word such as D300 or a group of 16 bit devices e.g. K4X000.

d) Values from 0 to 32,767 may be used in the data table.

e) The ABSD instruction may only be used **ONCE**.

From the example instruction and the data table below, the following timing diagram for elements M0 to M3 can be constructed.

When counter S2 equals the value below, the destination device D is		Assigned destination device D
turned ON	turned OFF	
D300 - 40	D301 - 140	M0
D302 - 100	D303 - 200	M1
D304 - 160	D305 - 60	M2
D306 - 240	D307 - 280	M3

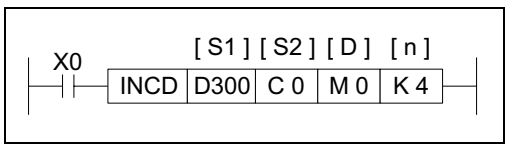


5.7.4 INCD (FNC 63)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S_1	S_2	D	n	
INCD FNC 63 (Incremental drum sequencer)	Generates a single output sequence in response to counter data	KnX, KnY, KnM, KnS, (in groups of 8) T, C, D	C Uses 2 consecutive counters	Y, M, S Note: n consecutive devices are used	K,H ⊠ Note: n≤ 64	INCD: 9 steps
		Note: High speed counters are not allowed				

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Operation Complete M8029



Operation:

This instruction generates a sequence of sequential output patterns (there are n number of addressed outputs) in response to the current value of a pair of selected counters (S2, S2+1).

Points to note:

a) This instruction uses a 'data table' which contains a single list of values which are to be selected and compared by two consecutive counters (S2and S2+1). The data table is identified as having a head address S1and consists of n data elements.

b) Counter S2 is programmed in a conventional way. The set value for counter S2 MUST be greater than any of the values entered into the data table. Counter S2 counts a user event and compares this to the value of the currently selected data element from the data table. When the counter and data value are equal, S2 increments the count of counter S2+1and resets its own current value to '0' (zero).
This new value of counter S2+1selects the new data element from the data table and counter S2now compares against the new data elements value.

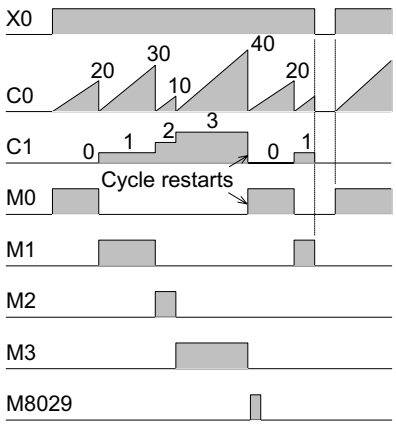
c) The counter S2+1 may have values from 0 to n. Once the nth data element has been processed, the operation complete flag M8029 is turned ON. This then automatically resets counter S2+1hence, the cycle starts again with data element S1+0.

d) Values from 0 to 32,767 may be used in the data table.

e) The INCD instruction may only be used **ONCE** in a program.

From the example instruction and the data table identified left, the following timing diagram for elements M0 to M3 can be constructed.

Data table		Value of counter S2+1
Data element	Data value / count value for counter S2	
D300	20	0
D301	30	1
D302	10	2
D303	40	3

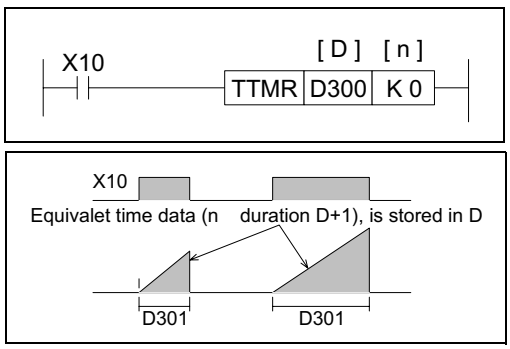


5.7.5 TTMR (FNC 64)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		D	n	
TTMR FNC 64 (Teaching timer)	Monitors the duration of a signal and places the timed data into a data register	D Note: 2 devices 16 bit words are used D and D+1	K, H ⊠ Note: n= 0: (D) = (D+1) × 1 n= 1: (D) = (D+1) × 10 n= 2: (D) = (D+1) × 100	TTMR: 5 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

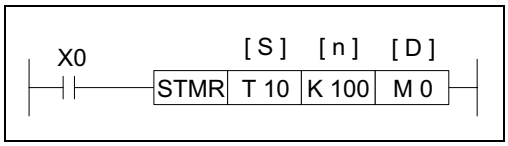
The duration of time that the TTMR instruction is energized, is measured and stored in device D+1 (as a count of 100ms periods).
The data value of D+1 (in secs), multiplied by the factor selected by the operand n, is moved in to register D. The contents of D could be used as the source data for an indirect timer setting or even as raw data for manipulation.
When the TTMR instruction is de-energized D+1 is automatically reset (D is unchanged).

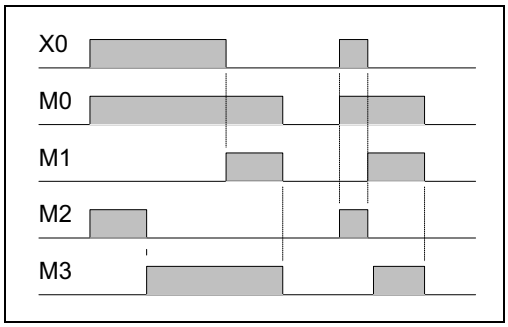
5.7.6 STMR (FNC 65)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	n	D	
STMR FNC 65 (Special timer)	Provides dedicated off-delay, one shot and flash timers	T Note: Timers 0 to 199 (100msec devices)	K, H ✉ Note: n= 1 to 32,767	Y, M, S Note:uses 4 consecutive devices D+0to D+3	STMR: 7 steps

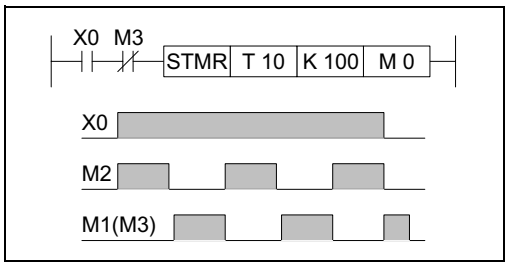
16 BIT OPERATION	32 BIT OPERATION	PULSE-P





Operation:

The designated timer Swill operate for the duration n with the operational effect being flagged by devices D+0to D+3. Device D+0is an off-delay timer, D+1is a one shot timer. When D+3 is used in the configuration below, D+1and D+2act in a alternate flashing sequence.

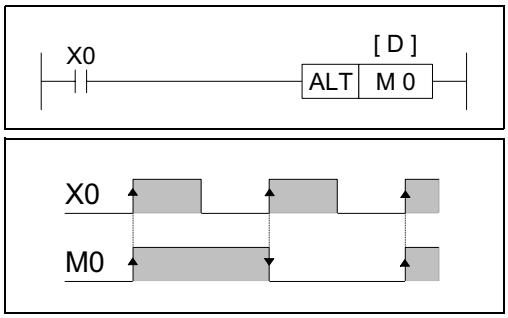


5.7.7 ALT (FNC 66)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		D	
ALT FNC 66 (Alternate state) ✈	The status of the assigned device is inverted on every operation of the instruction	Y, M, S	ALT, ALTP: 3 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The status of the destination device (D) is alternated on every operation of the ALT instruction.

This means the status of each bit device will flip-flop between ON and OFF. This will occur on every program scan unless a pulse modifier or a program interlock is used.

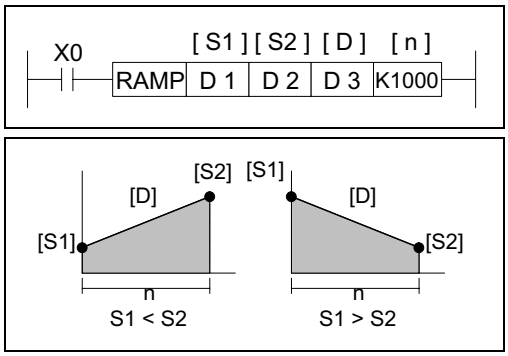
The ALT instruction is ideal for switching between two modes of operation e.g. start and stop, on and off etc.

5.7.8 RAMP (FNC 67)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S1	S2	D	n	
RAMP FNC 67 (Ramp variable value)	Ramps a device from one value to another in the specified number of steps	D Note: Device D uses two consecutive registers identified as D and D+1 these are read only devices.			K, H ⊠ Note: n= 1 to 32,767	RAMP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Operation Complete M8029

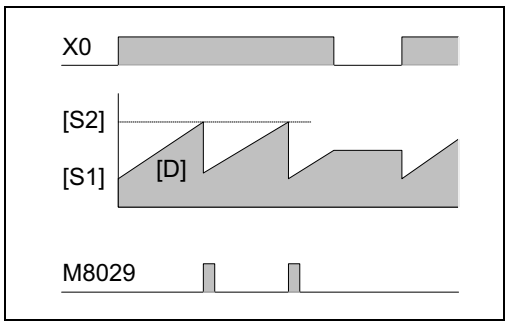


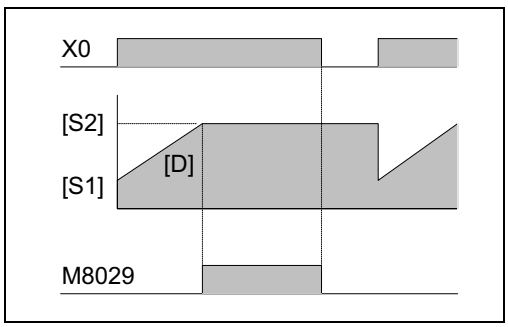
Operation:

The RAMP instruction varies a current value (D) between the data limits set by the user (S1and S2). The 'journey' between these extreme limits takes n program scans. The current scan number is stored in device D+1. Once the current value of D equals the set value of S2the execution complete flag M8029 is set ON.
The RAMP instruction can vary both increasing and decreasing differences between S1and S2.

Points to note:

a) FX2N and FX2NC users may set the operation mode of the RAMP instruction by controlling the state of special auxiliary relay M8026. When M8026 is OFF, the RAMP instruction will be in repeat mode. This means when the current value of D equals S2 the RAMP instruction will automatically reset and start again, i.e. the contents of D will be reset to that of S1 and the device D+1 (the number of current scans) will reset to '0' (zero). This is shown in the diagram opposite.
When M8026 is set ON, users will be operating the RAMP instruction in 'Hold mode'. This





means once the current value of D equals that of S2, the RAMP instruction will 'freeze' in this state. This means the M8029 will be set ON for as long as the instruction remains energized and the value of D will not reset until the instruction is re-initialized, i.e. the RAMP instruction is turned from OFF to ON again.

b) Users of FX_{1N} and FX_{1S} PLC's cannot change the operating mode of the RAMP instruction. For these PLC's the mode is fixed as in the same case as FX PLC's when M8026 has been set ON, i.e. HOLD mode.

c) If the RAMP instruction is interrupted before completion, then the current position within the ramp is 'frozen' until the drive signal is re-established. Once the RAMP instruction is re-driven registers D and D+1 reset and the cycle starts from its beginning again.

d) If the RAMP instruction is operated with a constant scan mode, i.e. D8039 is written to with the desired scan time (slightly longer than the current scan time) and M8039 is set ON. This would then allow the number of scans n (used to create the ramp between S1and S2) to be associated to a time. If 1 scan is equal to the contents of D8039 then the time to complete the ramp is equal to n $\times$ D8039.

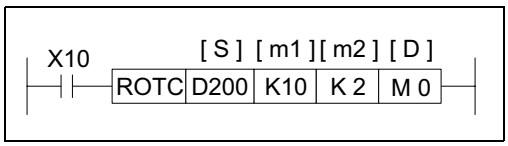
The RAMP instruction may also be used with special M flags M8193 and M8194 to mimic the operation of the SER (FNC 61) and RS (FNC 80) respectively when being programmed on older versions of programming peripherals. See page 1-5 for more details.

5.7.9 ROTC (FNC 68)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S	m1	m2	D	
ROTC FNC 68 (Rotary table control)	Controls a rotary tables movement is response to a requested destination/ position	D Note: uses 3 consecu-tive devices S+1≤ m1	K, H ⊠ Note: m1= 2 to 32,767	K, H ⊠ Note: m2= 0 to 32,767	Y, M, S Note: uses 8 consecu-tive devices	ROTC: 9 steps
			m1≥ m2			

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

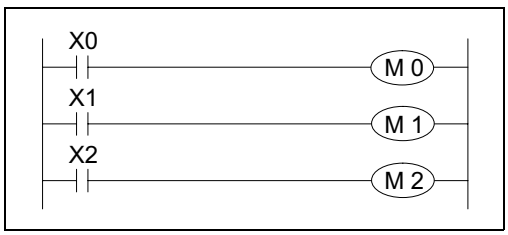
The ROTC instruction is used to aid the tracking and positional movement of the rotary table as it moves to a specified destination.

Points to note:

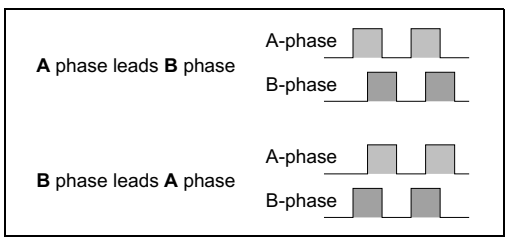
a) This instruction has many automatically defined devices. These are listed on the right of this page.

b) The ROTC instruction may only be used **ONCE**.

c) The ROTC instruction uses a built in 2-phase counter to detect both movement direction and distance travelled. Devices D+0and D+1 are used to input the phase pulses, while device D+2is used to input the 'zero position' on the rotary table. These devices should be programmed as shown in the example below (where the physical termination takes place at the associated X inputs).



The movement direction is found by checking the relationship of the two phases of the 2 phase counter, e.g.



Assigned devices

Indirect user selected devices:

D+0 A-phase counter signal - input
D+1 B-phase counter signal - input
D+2 Zero point detection - input
D+3 High speed forward - output
D+4 Low speed forward - output
D+5 Stop - output
D+6 Low speed reverse - output
D+7 High speed reverse - output

Rotary table constants:

m1 Number of encoder pulses per table revolution
m2 Distance to be travelled at low speed (in encoder pulses)

Operation variables:

S+0 Current position at the 'zero point' READ ONLY
S+1 Destination position (selected station to be moved to) relative to the 'zero point' - User defined
S+2 Start position (selected station to be moved) relative to the 'zero point' -User defined

d) When the 'zero point' input (D+2) is received the contents of device S+0 is reset to '0' (zero). Before starting any new operation it is advisable to ensure the rotary table is initialized by moving the 'zero point' drive dog or marker around to the 'zero point' sensor. This could be considered as a calibration technique. The re-calibration of the rotary table should be carried out periodically to ensure a consistent/accurate operation.

e) Devices D+3 to D+7 are automatically set by the ROTC instruction during its operation. These are used as flags to indicate the operation which should be carried out next.

f) All positions are entered in the form of the required encoder pulses. This can be seen in the following example:

- Example:

A rotary table has an encoder which outputs 400 (m1) pulses per revolution. There are 8 stations (0 to 7) on the rotary table. This means that when the rotary table moves from one station to its immediately following station, 50 encoder pulses are counted. The 'zero position' is station '0' (zero). To move the item located at station 7 to station 3 the following values must be written to the ROTC

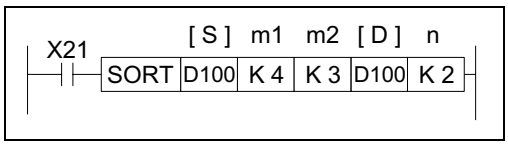
instruction:
S+1=3 $\times$ 50 = 150 (station 3's position in encoder pulses from the zero point)
S+2=7 $\times$ 50 = 350 (station 7's position in encoder pulses from the zero point)
m1= 400 (total number of encoder pulses per rev)
The rotary table is required approach the destination station at a slow speed starting from 1.5 stations before the destination. Therefore;
m2= 1.5 $\times$ 50 = 75 slow speed distance either side of the destination station (in encoder pulses)

5.7.10 SORT (FNC 69)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands					Program steps
		S	m1	m2	D	n	
SORT FNC 69 (SORT Tabulated Data)	Data in a defined table can be sorted on selected fields while retaining record integrity	D ⊠	K, H ⊠ Note: m1= 1 to 32 m2= 1 to 6		D ⊠	K, H D ⊠ Note: n = 1 to m2	SORT: 11 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Operation Complete M8029



Operation:

This instruction constructs a data table of m_1 records with m_2 fields having a start or head address of S. Then the data in field nis sorted in to numerical order while retaining each individual records integrity. The resulting (new) data table is stored from destination device D.

Points to note:

a) When a sort occurs each record is sorted in to ascending order based on the data in the selected sort field n.

b) The source (S) and destination (D) areas can be the same BUT if the areas are chosen to be different, there should be no overlap between the areas occupied by the tables.

c) Once the SORT operation has been completed the 'Operation Complete Flag' M8029 is turned ON. For the complete sort of a data table the SORT instruction will be processed m1times.

d) During a SORT operation, the data in the SORT table must not be changed. If the data is changed, this may result in an incorrectly sorted table.

e) The SORT instruction may only be used **ONCE** in a program.

From the example instruction and the 'data table' below left, the following data manipulation will occur when 'n' is set to the identified field

Original

		FIELD (m2)		
		1	2	3
RECORD (m1)	1	(D100) 32	(D104) 162	(D108) 4
	2	(D101) 74	(D105) 6	(D109) 200
	3	(D102) 100	(D106) 80	(D110) 62
	4	(D103) 7	(D107) 34	(D111) 6

Table1st table sort when n= 2

		FIELD (m2)		
		1	2	3
RECORD (m1)	1	(D100) 74	(D104) 6	(D108) 200
	2	(D101) 7	(D105) 34	(D109) 6
	3	(D102) 100	(D106) 80	(D110) 62
	4	(D103) 32	(D107) 162	(D111) 4

2nd table sort when n=1

		FIELD (m2)		
		1	2	3
RECORD (m1)	1	(D100) 7	(D104) 34	(D108) 6
	2	(D101) 32	(D105) 162	(D109) 4
	3	(D102) 74	(D106) 6	(D110) 200
	4	(D103) 100	(D107) 80	(D111) 62

MEMO

Applied Instructions:

FX1S | FX1N | FX2N | FX2NC

5.8 External FX I/O Devices - Functions 70 to 79

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

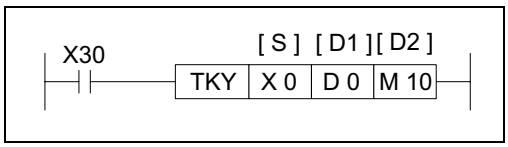
✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.8.1 TKY (FNC 70)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D1	D2	
TKY FNC 70 (Ten key input)	Reads 10 devices with associated decimal values into a single number	X, Y, M, S Note: uses 10 consecutive devices (identified as S_{+0} to S_{+9})	KnY, KnM, KnS, T, C, D, V, Z Note: uses 2 consecutive devices for 32 bit operation	Y, M, S Note: uses 11 consecutive devices (identified D_{2+0} to D_{2+10})	TKY: 7 steps DTKY: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

This instruction can read from 10 consecutive devices(S_{+0} to S_{+9}) and will store an entered numeric string in device D_1.

Points to note:

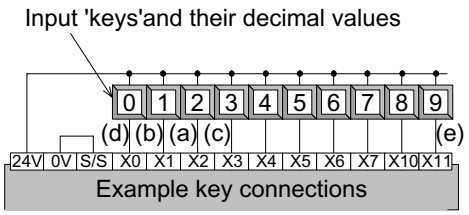
a) When a source device becomes active its associated destination (bit) device D_2 also becomes active. This destination device will remain active until another source device is operated. Each source device maps directly to its own D_2 device, i.e. S_{+0} maps to D_{2+0}, S_{+7} maps to D_{2+7} etc. These in turn, map directly to decimal values which are then stored in the destination data devices specified by D_1.

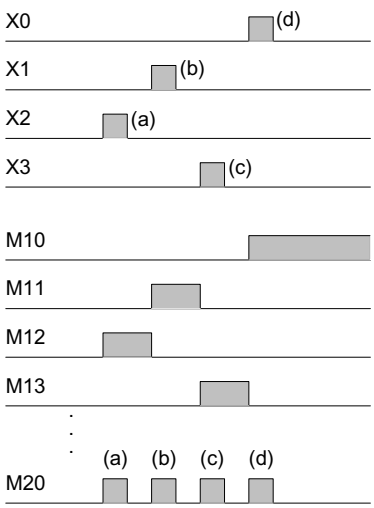
b) One source device may be active at any one time. The destination device D_{2+10} is used to signify that a key (one of the 10 source devices) has been pressed. D_{2+10} will remain active for as long as the key is held down. When the TKY instruction is active, every press of a key adds that digit to the stored number in D_1. When the TKY is OFF, all of the D_2 devices are reset, but the data value in D_1 remains intact.

c) When the TKY instruction is used with 16 bit operation, D_1 can store numbers from 0 to 9,999 i.e. max. 4 digits. When the DTKY instruction is used (32 bit operation) values of 0 to 99,999,999 (max. 8 digits) can be accommodated in two consecutive devices $D_1$and D_{1+1}. In both cases if the number to be stored exceeds the allowable ranges, the highest digits will overflow until an allowable number is reached. The overflowed digits are lost and can no longer be accessed by the user. Leading zero's are not accommodated, i.e. 0127 will actually be stored as 127 only.

d) The TKY instruction may only be used **ONCE**.

e) Using the above instruction as a brief example: If the 'keys' identified (a) to (d) are pressed in that order the number 2,130 will be entered into D_1. If the key identified as (e) is then pressed the value in D_1 will become 1,309. The initial '2' has been lost.



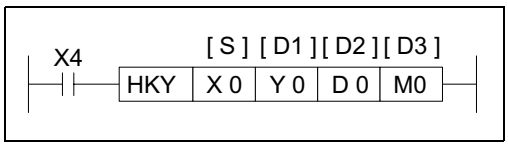


5.8.2 HKY (FNC 71)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S	D1	D2	D3	
HKY FNC 71 (Hexadeci-mal key input)	Multiplexes inputs and outputs to create a numeric keyboard with 6 function keys	X, Note: uses 4 consecu-tive devices	Y, Note: uses 4 consecu-tive devices	T, C, D, V, Z Note: uses 2 consecutive devices for 32 bit operation	Y, M, S Note: uses 8 consecu-tive devices	HKY: 9 steps DHKY: 17 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Operation Complete M8029



Operation 1 - Standard:

This instruction creates a multiplex of 4 outputs (D_1) and 4 inputs (S) to read in 16 different devices. Decimal values of 0 to 9 can be stored while 6 further function flags may be set.

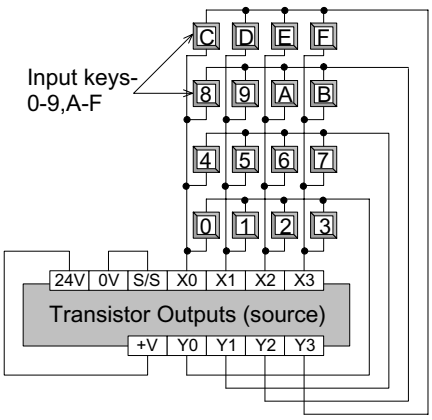
Points to note:

a) Each of the first 10 multiplexed source devices (identified as 0 to 9) map directly to decimal values 0 to 9. When entered, i.e. a source device is activated, then its associated decimal value is added to the data string currently stored in D_2. Activation of any of these keys causes bit device D_{3+7} to turn ON for the duration of that key press.

b) The last 6 multiplexed source devices (identified as function keys A to F) are used to set bit devices D_{3+0} to D_{3+5} respectively. These bit flags, once set ON, remain ON until the next function key has been activated. Activation of any of these keys causes bit device D_{3+6} to turn ON for the duration of that key press.

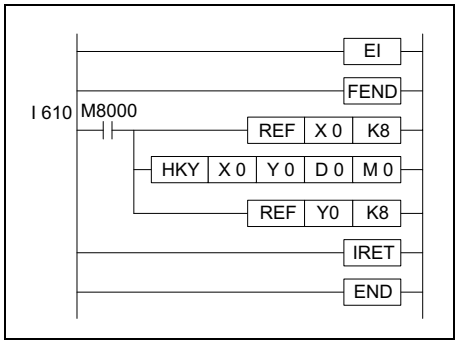
c) In all key entry cases, when two or more keys are pressed, only the key activated first is effective. When the pressing of a key is sensed the M8029 (execution complete flag) is turned ON. When the HKY instruction is OFF, all D_3 devices are reset but data value D_2 remains intact.

d) When the HKY instruction is used with 16 bit operation, D_2 can store numbers from 0 to 9,999 i.e. max. 4 digits. When the DHKY instruction is used (32 bit operation) values of 0 to 99,999,999 (max. 8 digits) can be accommodated in two consecutive devices D_2 and D_{2+1}. In both cases if the number to be stored exceeds the allowable ranges, the highest digits will overflow until an allowable number is reached. The over-flowed digits are lost and can no longer be accessed by the user. Leading zero's are not accommodated, i.e. 0127 will actually be stored as 127 only. This operation is similar to that of the TKY instruction.



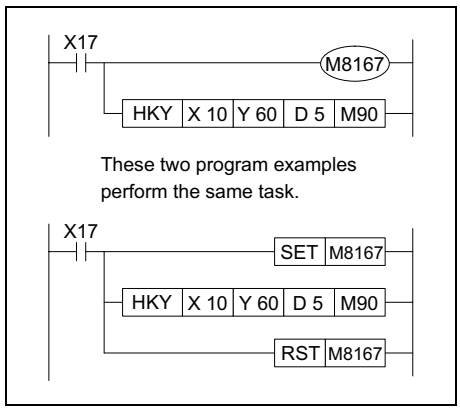
e) The HKY instruction may only be used **ONCE**.

f) Normal operation requires 8 scans to read the key inputs. To achieve a steady and repeatable performance, constant scan mode should be used, i.e. M8039 is set ON and a user defined scan time is written to register D8039. However, for a faster response the HKY instruction should be programmed in a timer interrupt routine as shown in the example opposite.



Operation 2 - Using the HKY Instruction With M8167:

(Applicable units: FX_{2N} and FX_{2NC})
When the HKY instruction is used with flag M8167 ON (as shown right), the operation of keys A through F allow actual entry of the Hexadecimal values of A through F respectively into the data device D_2. This is in addition to the standard 0 through 9 keys. All other operation is as specified in 'Operation 1 - Standard'. Maximum storage values for this operation become FFFF in 16 bit mode and FFFFFFFF in 32 bit (double word) mode.

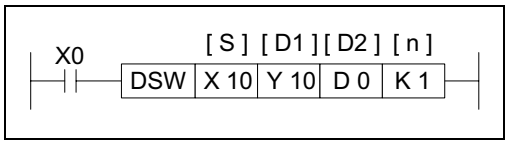


5.8.3 DSW (FNC 72)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S	D1	D2	n	
DSW FNC 72 (Digital switch)	Multiplexed reading of n sets of digital (BCD) thumbwheels	X Note: If n=2 then 8 devices else 4.	Y Note: uses 4 consecutive devices	T, C, D, V, Z Note: If n=2 then 2 devices else 1.	K, H ✉ Note: n= 1 or 2	DSW: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Operation Complete M8029



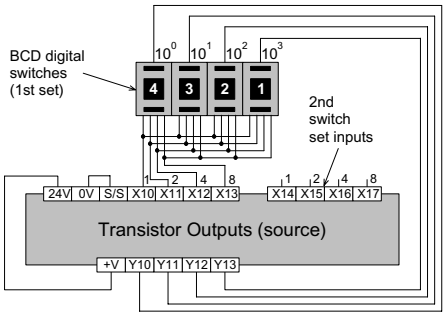
Operation:

This instruction multiplexes 4 outputs (D_1) through 1 or 2(n) sets of switches. Each set of switches consists of 4 thumbwheels providing a single digit input.

Points to note:

a) When n = 1 only one set of switches are read. The multiplex is completed by wiring the thumbwheels in parallel back to 4 consecutive inputs from the head address specified in operand S. The (4 digit) data read is stored in data device D_2.

Continued on next page...



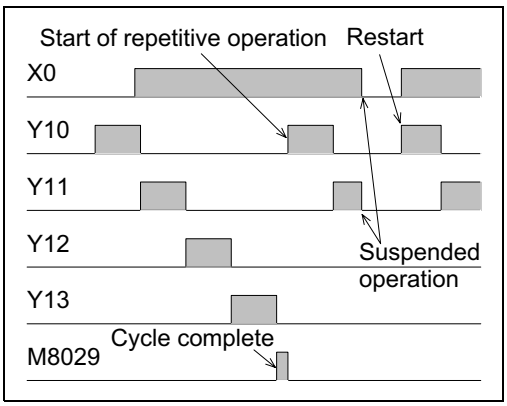
b) When n= 2, two sets of switches are read. This configuration requires 8 consecutive inputs taken from the head address specified in operand S. The data from the first set of switches, i.e. those using the first 4 inputs, is read into data device D_2. The data from the second set of switches (again 4 digits) is read into data device D_{2+1}.

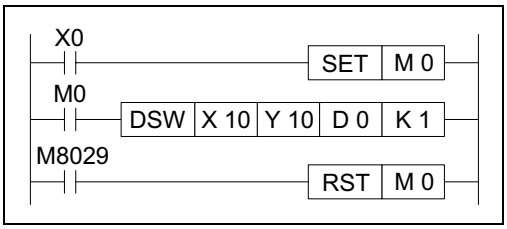
c) The outputs used for multiplexing (D_1) are cycled for as long as the DSW instruction is driven. After the completion of one reading, the execution complete flag M8029 is set. The number of outputs used does **not** depend on the number of switches n.

d) If the DSW instruction is suspended during mid-operation, when it is restarted it will start from the beginning of its cycle and not from its last status achieved.

e) It is recommended that transistor output units are used with this instruction. However, if the program technique at the right is used, relay output units can be successfully operated as the outputs will not be continually active.

f) The DSW instruction may be used **TWICE** on FX_{2N} & FX_{2NC} controllers. FX_{1S} & FX_{1N} units can operate an **Unlimited** number of DSW instructions.



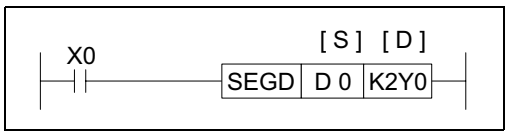


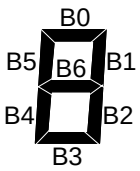
5.8.4 SEGD (FNC 73)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
SEGD FNC 73 (Seven segment decoder)	Hex data is decoded into a format used to drive seven segment displays	K, H KnX, KnY, KnM, KnS, T, C, D, V, Z Note: Uses only the lower 4 bits	KnY, KnM, KnS, T, C, D, V, Z Note: The upper 8 bits remain unchanged	SEGD, SEGDP: 5 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Zero M8020





It can be seen that B7 is NOT used. Hence B7 of the destination device D will always be OFF,

Operation:

A single hexadecimal digit (0 to 9, A to F) occupying the lower 4 bits of source device S is decoded into a data format used to drive a seven segment display. A representation of the hex digit is then displayed. The decoded data is stored in the lower 8 bits of destination device D. The upper 8 bits of the same device are not written to. The diagram opposite shows the bit control of the seven segment display. The active bits correspond to those set to 1 in the lower 8 bits of the destination device D.

5.8.5 SEGL (FNC 74)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D	n	
SEGL FNC 74 (Seven segment with latch)	Writes data to multiplexed single digit displays - 4 digits per set, max. 2 sets	K, H KnX, KnY, KnM, KnS T, C, D, V, Z	Y Note: n = 0 to 3, 8 outputs are used n = 4 to 7, 12 outputs are used	K, H, ☒ Note: n= 0 to 3, 1 set of 7 Seg active n= 4 to 7, 2 sets of 7 Seg active	SEGL: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Operation Complete M8029

Operation:

This instruction takes a source decimal value (S) and writes it to a set of 4 multiplexed, outputs (D). Because the logic used with latched seven segment displays varies between display manufactures, this instruction can be modified to suit most logic requirements. Configurations are selected depending on the value of n, see the following page.

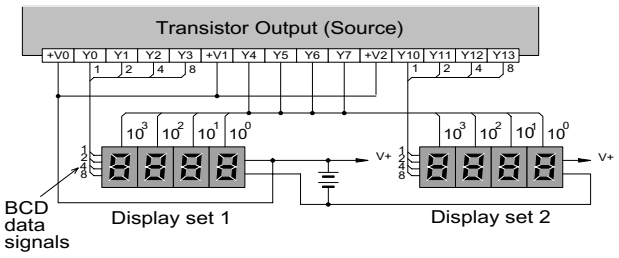
Points to note:

a) Data is written to a set of multiplexed outputs (D_{+0} to D_{+7}, 8 outputs) and hence seven segment displays. A set of displays consists of 4 single digit seven segment units. A maximum of two sets of displays can be driven with this instruction. When two sets are used the displays share the same strobe outputs (D_{+4} to D_{+7} are the strobe outputs). An additional set of 4 output devices is required to supply the new data for the second set of displays (D_{+10} to D_{+13}, this is an octal addition). The strobe outputs cause the written data to be latched at the seven segment display.

b) Source data within the range of 0 to 9,999 (decimal) is written to the multiplexed outputs. When one set of displays are used this data is taken from the device specified as operand S. When two sets of displays are active the source device S_{+1} supplies the data for the second set of displays. This data must again be within the range 0 to 9,999. When using two sets of displays the data is treated as **two** separate numbers and is **not** combined to provide a single output of 0 to 99,999,999.

c) The SEGL instruction takes 12 program scans to complete one output cycle regardless of the number of display sets used. On completion, the execution complete flag M8029 is set.

Note: A single set of strobe signals are always used regardless of the number of display sets.



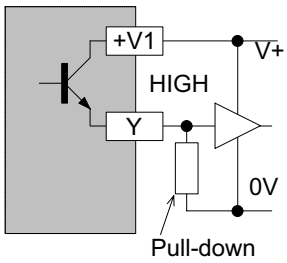
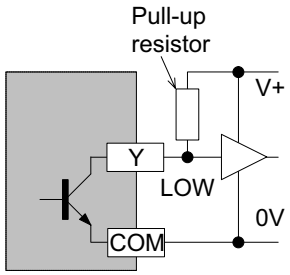
In this example it has been assumed that the seven segment displays accept data HIGH inputs and latch when a HIGH signal is received

d) If the SEGL instruction is suspended during mid-operation, when it is restarted it will start from the beginning of its cycle and not from its last status achieved.

e) The SEGL instruction may be used **TWICE** on FX2N & FX2NC controllers. FX1S & FX1N units can operate an **Unlimited** number of SEGL instructions.

Selecting the correct value for operand n

The selection of parameter n depends on 4 factors;
1) The logic type used for the PLC output
2) The logic type used for the seven segment data lines
3) The logic type used for the seven segment strobe signal

Device considered		Positive logic	Negative logic
PLC Logic		Source output 	Sink output 
		With a source output, when the output is HIGH the internal logic is '1'	With a sink output, when the output is LOW the internal logic is '1'
Seven segment Display logic	Strobe signal logic	Data is latched and held when this signal is HIGH, i.e. its logic is '1'	Data is latched and held when this signal is LOW, i.e. its logic is '1'
	Data signal logic	Active data lines are held HIGH, i.e. they have a logic value of '1'	Active data lines are held LOW, i.e. they have a logic value of '1'

There are two types of logic system available, positive logic and negative logic. Depending on the type of system, i.e. which elements have positive or negative logic the value of n can be selected from the table below with the final reference to the number of sets of seven segment displays being used:

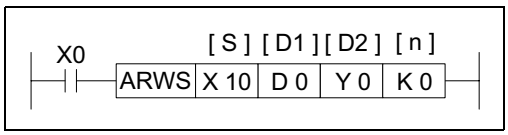
PLC Logic	Seven segment display logic		n	
	Data Logic	Strobe logic	1 display set	2 display sets
Positive (Source)	Positive (High)	Positive (High)	0	4
Negative (Sink)	Negative (Low)	Negative (Low)		
Positive (Source)	Positive (High)	Negative (Low)	1	5
Negative (Sink)	Negative (Low)	Positive (High)		
Positive (Source)	Positive (High)	Negative (Low)	2	6
Negative (Sink)	Negative (Low)	Positive (High)		
Positive (Source)	Positive (High)	Positive (High)	3	7
Negative (Sink)	Negative (Low)	Negative (Low)		

5.8.6 ARWS (FNC 75)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S	D1	D2	n	
ARWS FNC 75 (Arrow switch)	Creates a user defined, (4 key) numeric data entry panel	X, Y, M, S Note: uses 4 consecu-tive devices	T, C, D, V, Z Note: data is stored in a decimal format	Y Note: uses 8 consecu-tive devices	K, H ⊠ Note: n= 0 to 3,	ARWS: 9 steps

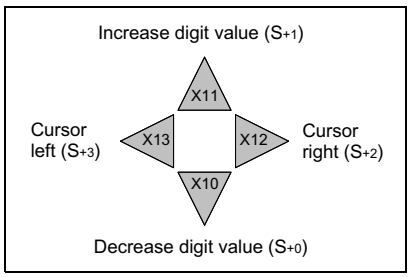
16 BIT OPERATION	32 BIT OPERATION	PULSE-P



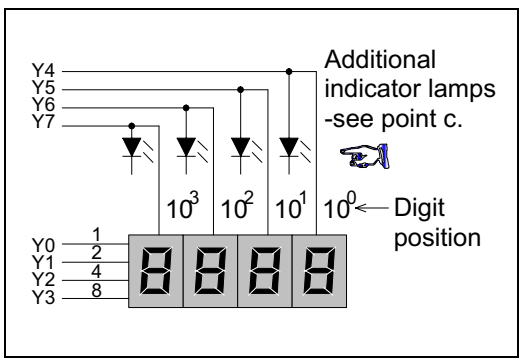
Operation:

This instruction displays the contents of a single data device D_1on a set of 4 digit, seven segment displays. The data within $D_1$is actually in a standard decimal format but is automatically converted to BCD for display on the seven segment units. Each digit of the displayed number can be selected and edited. The editing procedure directly changes the value of the device specified as D_1.

Points to note:



a) The data stored in destination device D_1can have a value from the range 0 to 9,999 (decimal), i.e. 4 digit data. Each digits data value, can be incremented (S_{+1}) or decremented (S_{+0}) by pressing the associated control keys. The edited numbers automatically 'wrap-around' from 9 - 0 - 1 and 1 -0 - 9. The digit data is displayed by the lower 4 devices from D_2, i.e. D_{2+0} to D_{2+3}.



b) On initial activation of the ARWS instruction, the digit in the numeric position 10^3 is currently selected. Each digit position can be sequentially 'cursored through' by moving to the left (S_{+2}) or to the right (S_{+3}). When the last digit is reached, the ARWS instruction automatically wraps the cursor position around, i.e. after position 10^3, position 10^0 is selected and vice-versa. Each digit is physically selected by a different 'strobe' output.

c) To aid the user of an operation panel controlled with the ARWS instruction, additional lamps could be wired in parallel with the strobe outputs for each digit. This would indicate which digit was currently selected for editing.

d) The parameter n has the same function as parameter n of the SEGL instruction - please see page5-86, 'Selecting the correct value for operand n'. Note: as the ARWS instruction only controls one set of displays only values of 0 to 3 are valid for n.

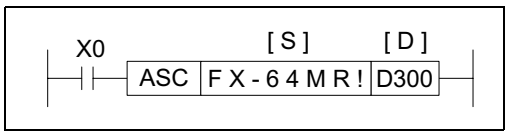
e) The ARWS instruction can be used **ONCE**. This instruction should only be used on transistor output PLC's.

5.8.7 ASC (FNC 76)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D1	
ASC FNC 76 (ASCII code conversion)	An entered alphanumeric string can be converted to its ASCII codes	Alphanumeric data e.g. 0-9, A - Z and a - z etc. Note: Only one, 8 character string may be entered at any one time.	T, C, D Note: uses 4 consecutive devices	ASC : 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The source data string S consists of up to 8 characters taken from the printable ASCII character (Char) set. If less than 8 Char are used, the difference is made up with null Char (ASCII 00).

The source data is converted to its associated ASCII codes. The codes are then stored in the destination devices D, see example shown below.

D	Byte	
	High	Low
D300	58 (X)	46 (F)
D301	36 (6)	2D (-)
D302	4D (M)	34 (4)
D303	21 (!)	52 (R)

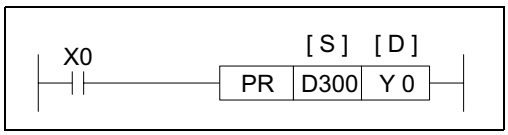
Note: ASCII Char **cannot** be entered from a hand held programmer.

5.8.8 PR (FNC 77)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D1	
PR FNC 77 (Print)	Outputs ASCII data to items such as display units	T, C, D Note: 8 byte mode (M8027=OFF) uses 4 consecutive devices 16 byte mode (M8027= ON) uses 8 consecutive devices	Y Note: uses 10 consecutive devices.	PR: 5 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Operation Complete M8029



Operation:

Source data (stored as ASCII values) is read byte by byte from the source data devices. Each byte is mapped directly to the first 8 consecutive destination devices D_{+0} to D_{+7}). The final two destination bits provide a strobe signal (D_{+10}, numbered in octal) and an execution/busy flag (D_{+11}, in octal).

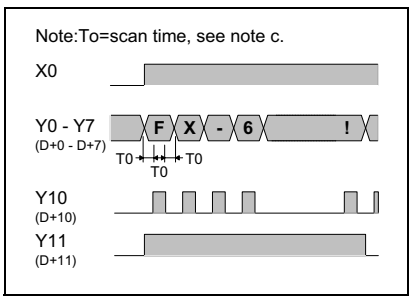
Points to note:

a) The source byte-data maps the lowest bit to the first destination device D_{+0}. Consequently the highest bit of the byte is sent to destination device D_{+7}.

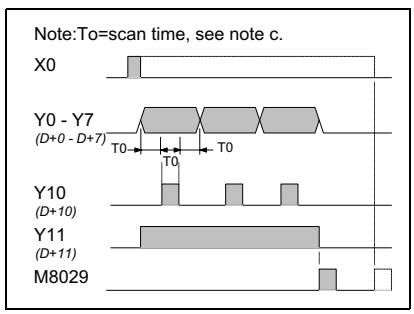
b) The PR instruction may only be used **TWICE** in a sequence program. This instruction should only be used on transistor output PLC's. The PR instruction will not automatically repeat its operation unless the drive input has been turned OFF and ON again.

c) The operation of the PR instruction is program scan dependent. Under standard circumstances it takes 3 program scans to send 1 byte. However, for a faster operation the PR instruction could be written into a timer interrupt routine similar to the one demonstrated for HKY on page 5-82.

d)8 byte operation has the following timing diagram. It should be noted that when the drive input (in the example X0) is switched OFF the PR instruction will cease operation. When it is restarted the PR instruction will start from the beginning of the message string. Once all 8 bytes have been sent the execution/busy flag is dropped and the PR instruction suspends operation.



e) 16 byte operation requires the special auxiliary flag M8027 to be driven ON (it is recommended that M8000 is used as a drive input). In this operation mode the drive input (in the example X0) does not have to be active all of the time. Once the PR instruction is activated it will operate continuously until all 16 bytes of data have been sent or the value 00H (null) has been sent. Once the operation is complete the execution/busy flag (D_{+11}, octal) is turned OFF and M8029 the execution complete flag is set.

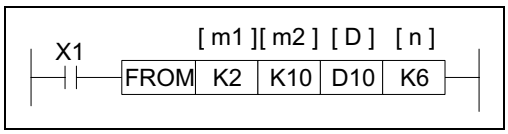


5.8.9 FROM (FNC 78)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		m1	m2	D	n	
FROM FNC 78 (FROM)	Read data from the buffer memories of attached special function blocks	K, H ⊠ Note: m1= 0 to 7	K, H ⊠ Note: m2 = FX(2C) 0 to 31, FX2N 0 to 32767	KnY, KnM, KnS, T, C, D, V, Z	K, H ⊠ Note: 16 bit op: n= 1 to 32 32 bit op: n= 1 to 16	FROM, FROMP: 9 steps DFROM, DFROMP: 17 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P

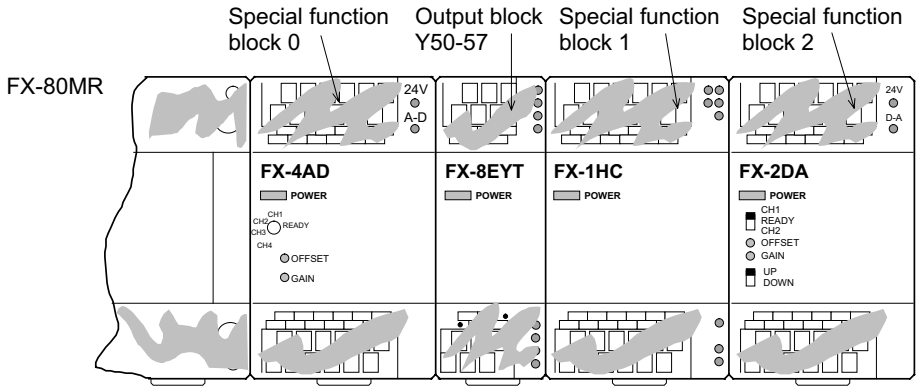


Operation:

The FROM instruction reads n words of data starting from the buffer memory address m2 of the special function block with the logical block position specified as m1, The read data is stored in the PLC at head address D for n word devices.

Points to note:

a) All special function blocks which are addressable with the FROM/TO instructions are connected to the extension bus on the right hand side of the PLC. Each special function block can be inserted at any point within the chain of extended units (as long as the system configuration rules are not broken). Each special function block is consecutively addressed from 0 to 7 beginning with the one closest to the base unit.



b) Each special function unit has different buffer memory registers. These often have a dedicated use for each individual unit. Before any reading or writing of data is undertaken ensure that the correct buffer memory allocations for the unit used are known.
m2: This defines the head address of the (special function blocks) buffer memories being accessed. m2 may have a value from the range 0 to 31.
n: This identifies the number of words which are to be transferred between the special function block and the PLC base unit. n may have a value of 1 to 31 for 16 bit operation but a range of 1 to 16 is available for 32 bit operation.

c) The destination head address for the data read FROM the special function block is specified under the D operand; and will occupy n further devices.

d) This instruction will only operate when the drive input is energized.

e) Users of all PLC models have the option of allowing interrupts to occur immediately, i.e. during the operation of the FROM/TO instructions or to wait until the completion of the current FROM/TO instruction. This is achieved by controlling the special auxiliary flag M8028. The following table identifies certain points associated with this control and operation.

Interruption Disabled	Interruption Enabled
M8028 = OFF	M8028 = ON
Jumps called by interrupt operation are delayed until the completion of the data transfer of the FROM/TO instruction	Jumps called by interrupt operation occur immediately
A small delay of (800m +200) μsec can be expected in the worst case. Note: m = the number of 32 bit words	Data transfer will resume upon return from the interrupt program. This may not be desirable if a FROM/TO instruction has been programmed within the called interrupt
Ensures that FROM/TO instructions included in an interrupt program will not interact with others elsewhere	M8028 should only be used when a very short delay is required in applications where timing and accuracy's are important

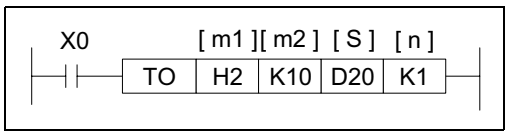
Users of FX1S have no option for interruption of the FROM/TO instructions and hence always operate in a mode equivalent to having M8028 switched OFF.

5.8.10 TO (FNC 79)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		m1	m2	S	n	
TO FNC 79 (TO)	Writes data to the buffer memories of attached special function blocks	K, H ⊠ Note: m1= 0 to 7	K, H ⊠ Note: m2 = 0 to 32767	K,H, KnX, KnY, KnM, KnS, T, C, D, V, Z	K, H ⊠ Note: 16 bit op: n= 1 to 32 32 bit op: n= 1 to 16	TO, TOP: 9 steps DTO, DTOP: 17 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The TO instruction writes n words of data to the head buffer memory address m2 of the special function block with the logical block position specified in m1. The written data is taken from the PLC's head address S for n word devices.

Points to note:

All points are the same as the FROM instruction (see previous page) except point c) which is replaced by the following:

a) The source head address for the data written TO the special function block is specified under the S operand.

MEMO

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.9 External FX Serial Devices - Functions 80 to 89

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

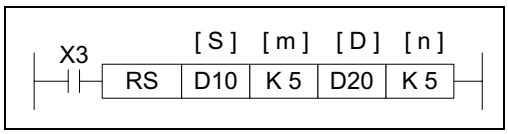
✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.9.1 RS (FNC 80)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S	m	D	n	
RS FNC 80 (Serial Communications instruction)	Used to control serial communications from/to the programmable controller	D (including file registers)	K, H, D ⊠ m = 1 to 256, FX2N 1 to 4096.	D	K, H, D ⊠ m = 1 to 256, FX2N 1 to 4096	RS: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Zero M8020 Borrow M8021 Carry M8022



Operation:

This instruction performs the direct control of communications over FX and FX0N communication adapters which connect to the left hand port of the Main Processing Unit, i.e. FX0N-232ADP, FX-232ADP etc.

Points to note:

a) This instruction has many automatically defined devices. These are listed in the boxed column to the right of this page.

b) The RS instruction has two parts, send (or transmission) and receive. The first elements of the RS instruction specify the transmission data buffer (S) as a head address, which contains m number of elements in a sequential stack.
The specification of the receive data area is contained in the last two parameters of the RS instruction. The destination (D) for received messages has a buffer or stack length of n data elements. The size of the send and receive buffers dictates how large a single message can be.
Buffer sizes may be updated at the following times:
1) Transmit buffer - before transmission occurs, i.e. before M8122 is set ON
2) Receive buffer - after a message has been received and before M8123 is reset.

c) Data cannot be sent while a message is being received, the transmission will be delayed - see M8121.

d) More than one RS instruction can be programmed but only one may be active at any one time.

e) Refer to the FX Communications Manual when using this function

Assigned devices

Data devices:

D8120 - Contains the configuration parameters for communication, i.e. Baud rate,Stop bits etc. Full details over the page

D8122 - Contains the current count of the number of remaining bytes to be sent in the currently transmitting message.

D8123 - Contains the current count of the number of received bytes in the 'incoming' message.

D8124 - Contains the ASCII code of the character used to signify a message header - default is 'STX', 02 HEX.

D8125 - Contains the ASCII code of the character used to signify a message terminator -default is 'ETX', 03 HEX.

Operational flags:

M8121 - This flag is ON to indicate a transmission is being delayed until the current receive operation is completed.

M8122 - This flag is used to trigger the transmission of data when it is set ON.

M8123 - This flag is used to identify (when ON) that a complete message has been received.

M8124 - Carrier detect flag. This flag is for use with FX and FX2C Main Processing Units. It is typically useful in modem communications

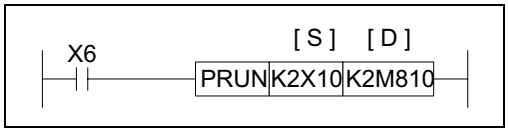
M8161 - 8 or 16 bit operation mode ON = 8 bit mode where only the lower 8 bits in each source or destination device are used, i.e. only one ASCII character is stored in one data register OFF = 16bit mode where all of the available source/ destination register is used, i.e. two ASCII characters are stored in each data register.

5.9.2 RUN (FNC 81)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
PRUN FNC 81 (Parallel run)	Used to control the FX parallel link adapters: FX2-40AW/AP	KnX, KnM	KnY, KnY	PRUN, PRUNP: 5 steps DPRUN, DPRUNP: 9 steps
		Note: n = 1 to 8 For ease and convenience, the head address bit should be a multiple of '10', e.g. X10, M100, Y30 etc.		

16 BIT OPERATION	32 BIT OPERATION	PULSE-P

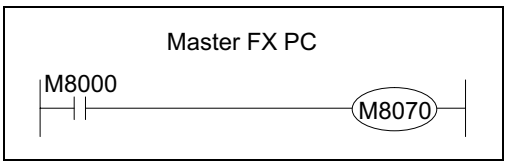


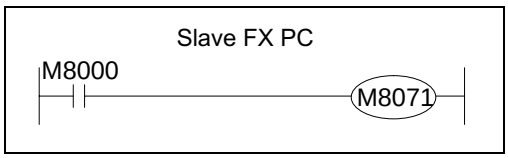
Operation:

This instruction is used with the FX parallel link adapters. It allows source data to be moved into the bit transmission area. The actual control of the parallel link communication is by special M flags.

Points to note:

a) Parallel link communications automatically take place when both systems are 'linked' and the Master station (M8070), Slave station flags (M8071) have been set ON (there is no need to have a PRUN instruction for communications). There can only be one of each type of station as this system connects only two FX PLC's. The programs shown opposite should be inserted into the appropriate FX PLC's programs.





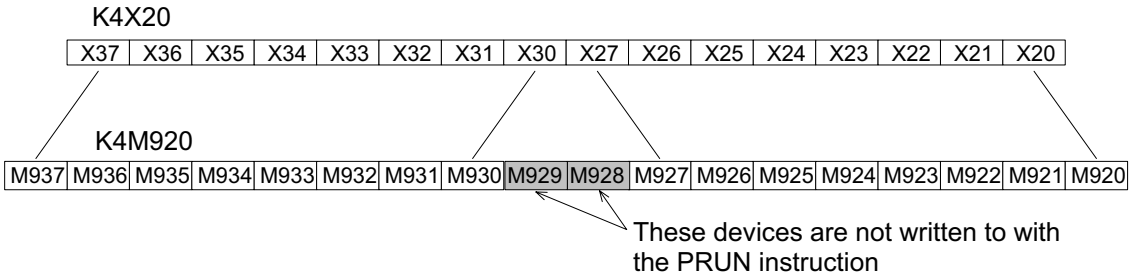
Once the station flags have been set, they can only be cleared by either forcibly resetting them when the FX PLC is in STOP mode or turning the power OFF and ON again.

b) During automatic communications the following data is 'swapped' between the Master and Slave PLC's.

Master station		Communication direction	Slave Station	
	Bit Data		Bit Data	
M8070 = ON	M800 to M899 (100 points)	→	M800 to M899 (100 points)	M8071 = ON
	M900 to M999 (100 points)	←	M900 to M999 (100 points)	
	Data words		Data words	
	D490 to D499 (10 points)	→	D490 to D499 (10 points)	
	D500 to D509 (10 points)	←	D500 to D509 (10 points)	

Continued...

c) The PRUN instruction enables data to be moved into the bit transmission area or out of the (bit) data received area. The PRUN instruction differs from the move statement in that it operates in octal. This means if K4X20 was moved using the PRUN instruction to K4M920, data would not be written to M928 and M929 as these devices fall outside of the octal counting system. This can be seen in the diagram below.



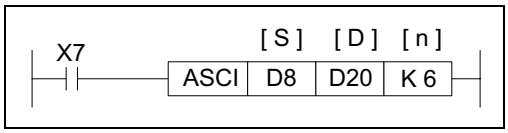
d) For more information please see page 9-6.

5.9.3 ASCI (FNC 82)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D	n	
ASCI FNC 82 (Converts HEX to ASCII)	Converts a data value from hexadecimal to ASCII	K, H, KnX, KnY, KnM, KnS T, C, D, V, Z	KnY, KnM, KnS T, C, D	K, H Note: n = 1 to 256 ✉	ASCI, ASCIP: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

This instruction reads n hexadecimal data characters from head source address (S) and converts them in to the equivalent ASCII code. This is then stored at the destination (D) for n number of bytes.

Points to note:

Please note that data is converted 'as read', i.e. using the example above with the following data in (D9,D8) AB**CD**H,**EF26**H. Taking the first n hexadecimal characters (digits) from the right (in this case n= 6) and converting them to ASCI will store values in 6 consecutive bytes from D20, i.e. D20 = (67, 68), D21 = (69, 70) and D22 = (50, 54) respectively. In true characters symbols that would be read as **CDEF26**.

This can be shown graphically as in the table to the right. Please take special note that the source data (S) read from the most significant device to the least significant. While the destination data (D) is read in the opposite direction.

Source (S)		Data
D9	b12-15	A
	b8-11	B
	b4-7	**C**
	b0-3	**D**
D8	b12-15	**E**
	b8-11	**F**
	b4-7	**2**
	b0-3	**6**

Destination (D)		ASCII Code HEX	ASCII Code DEC	Symbol
D20	b8-15	43	67	'C'
	b0-7	44	68	'D'
D21	b8-15	45	69	'E'
	b0-7	46	70	'F'
D22	b8-15	32	50	'2'
	b0-7	36	54	'6'

The ASCI instruction can be used with the M8161, 8 bit/16bit mode flag. The effect of this flag is exactly the same as that detailed on page 10-20. The example to the right shows the effect when M8161 is OFF.

If M8161 was set ON, then only the lower destination byte (b0-7) would be used to store data and hence 6 data registers would be required (D20 through D25).

ASCII Character Codes

The table below identifies the usable hexadecimal digits and their associated ASCII codes

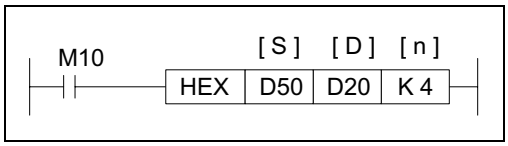
HEX Character		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ASCII Code	HEX	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46
	DEC	48	49	50	51	52	53	54	55	56	57	65	66	67	68	69	70
Character Symbol		'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'A'	'B'	'C'	'D'	'E'	'F'

5.9.4 HEX (FNC 83)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D	n	
HEX FNC 83 (Converts ASCII to HEX)	Converts a data value from ASCII in to a hexadecimal equivalent	K, H, KnX, KnY, KnM, KnS T, C, D	KnY, KnM, KnS T, C, D, V, Z	K, H Note: n = 1 to 256 ✉	HEX, HEXP: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

This instruction reads n ASCII data bytes from head source address (S) and converts them in to the equivalent Hexadecimal character. This is then stored at the destination (D) for n number of bytes.

Points to note:

Please note that this instruction 'works in reverse' to the ASCI instruction, i.e. ASCII data stored in bytes is converted into associated hexadecimal characters. The HEX instruction can be used with the M8161 8bit/16bit flag. In this case the source data (S)is read from either the lower byte (8bits) when M8161 is ON, or the whole word when M8161 is OFF i.e. using the example above with the following data in devices D50 and D51 respectively (43_H,41_H) (42_H,31_H) and assuming M8161 is ON.

The ASCII data is converted to its hexadecimal equivalent and stored sequentially digit by digit from the destination head address.
If M8161 had been OFF, then the contents of D20 would read $CAB1_H$.

Source (S)	ASCII Code HEX	ASCII Code DEC	Symbol
D51 b8-15	43	67	'C'
D51 b0-7	41	65	**'A'**
D50 b8-15	42	66	'B'
D50 b0-7	31	49	**'1'**

Destination (D)	Data
D20 b12-15	-
D20 b8-11	-
D20 b4-7	A
D20 b0-3	1

For further details regarding the use of the HEX instruction and about the available ASCII data ranges, please see the following information point 'ASCII Character Codes' under the ASCI instruction on the previous page.

Important:

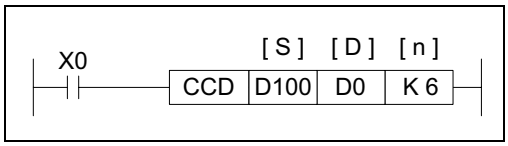
If an attempt is made to access an ASCII Code (HEX or Decimal) which falls outside of the ranges specified in the table on previous page, the instruction is not executed. Error 8067 is flagged in data register D8004 and error 6706 is identified in D8067. Care should be taken when using the M8161 flag, and additional in the specification of the number of element 'n' which are to be processed as these are the most likely places where this error will be caused.

5.9.5 CCD (FNC 84)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D	n	
CCD FNC 84 (Check Code)	Checks the 'vertical' parity of a data stack	KnX, KnY, KnM, KnS T, C, D	KnY, KnM, KnS T, C, D	K, H D Note: n = 1 to 256 ✉	CCD, CCDP: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

This instruction looks at a byte (8 bit) stack of data from head address (S)for n bytes and checks the vertical bit pattern for parity and sums the total data stack. These two pieces of data are then stored at the destination (D).

Points to note:

a) The SUM of the data stack is stored at destination D while the Parity for the data stack is stored at D_{+1}.

b) During the Parity check an even result is indicated by the use of a 0 (zero) while an odd parity is indicated by a 1 (one).

c) This instruction can be used with the 8 bit/ 16 bit mode flag M8161. The following results will occur under these circumstances. See page 10-20 for more details about M8161.

M8161=OFF									
Sourse (S)		Bit patterm							
D100 H	FF	1	1	1	1	1	1	1	1
D100 L	FF	1	1	1	1	1	1	1	1
D101 H	FF	1	1	1	1	1	1	1	1
D101 L	00	0	0	0	0	0	0	0	0
D102 H	F0	1	1	1	1	0	0	0	0
D102 L	0F	0	0	0	0	1	1	1	1
Vertical party D1		***0***	***0***	***0***	***0***	***0***	***0***	***0***	***0***
SUM D0		***3FC***							

M8161=ON									
Sourse (S)		Bit patterm							
D100 L	FF	1	1	1	1	1	1	1	1
D101 L	00	0	0	0	0	0	0	0	0
D102 L	0F	0	0	0	0	1	1	1	1
D103 L	F0	1	1	1	1	0	0	0	0
D104 L	F0	1	1	1	1	0	0	0	0
D105 L	0F	0	0	0	0	1	1	1	1
Vertical party D1		***1***	***1***	***1***	***1***	***1***	***1***	***1***	***1***
SUM D0		***2FD***							

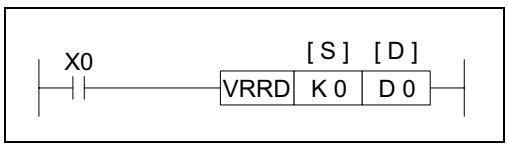
It should be noted that when M8161 is OFF 'n' represents the number of consecutive bytes checked by the CCD instruction. When M8161 is ON only the lower bytes of 'n' consecutive words are used.
The 'SUM' is quite simply a summation of the total quantity of data in the data stack. The Parity is checked vertically through the data stack as shown by the shaded areas.

5.9.6 VRRD (FNC 85)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
VRRD FNC 85 (Volume read)	Reads an analog value from 1 of 8 volume inputs on the FX-8AV	K, H Note: S= 0 to 7 corresponding to the 8 available volumes on the FX-8AV	KnY, KnM, KnS T, C, D, V, Z	VRRD, VRRDP: 5 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The identified volume (S) on the FX-8AV is read as an analog input. The analog data is in an 8 bit format, i.e. values from 0 to 255 are readable. The read data is stored at the destination device identified under operand D.

Note:

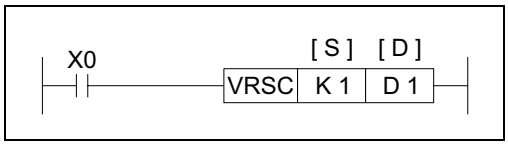
The FX-8AV volume 'inputs' are able to be read in two formats, a) as an analog value and b) as an 11 (0 to 10) position rotary switch. The second use is described in the VRSC instruction (FNC 86).

5.9.7 VRSD (FNC 86)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
VRSC FNC 86 (Volume scale)	Reads the set position value, 0 to 10, from volume inputs on the FX-8AV	K, H Note: S= 0 to 7 corresponding to the 8 available volumes on the FX-8AV	KnY, KnM, KnS T, C, D, V, Z	VRSC, VRSCP: 5 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The identified volume (S) on the FX-8AV is read as a rotary switch with 11 set positions (0 to 10). The position data is stored at device D as an integer from the range 0 to 10.

Note:

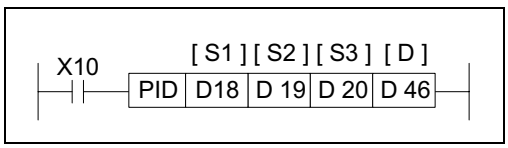
The FX-8AV volume 'inputs' are able to be read in two formats, a) as a 11 (0 to 10) position rotary switch and b) as an analog value. The second use is described in the VRRD instruction (FNC 85).

5.9.8 PID (FNC 88)

FX1S | FX1N | FX2N | FX2NC

Mnemonic	Function	Operands				Program steps
		S1	S2	S3	D	
PID FNC 88 (PID control loop) register each	Receives a data input and calculates a corrective action to a specified level based on PID control	D⊠ Note: S1 and S2 use a single data register		D⊠ Note: S3 uses 25 consecutive data registers	D⊠ Note: D uses a single data register	PID: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

This instruction takes a current value (S_2) and compares it to a predefined set value (S_1). The difference or error between the two values is then processed through a PID loop to produce a correction factor which also takes into account previous iterations and trends of the calculated error. The PID process calculates a correction factor which is applied to the current output value and stored as a corrected output value in destination device (D). The setup parameters for the PID control loop are stored in 25 consecutive data registers S_3+0 through S_3+24.

Points to note:

a) Every PID application is different. There will be a certain amount of “trial and error” necessary to set the variables at optimal levels.

b) On FX2N, FX2NC & FX1N MPUs a Pre-tuning feature is available that can quickly provide initial values for the PID process. Refer to page 10-28 for more details.

c) The FX1S does not have analog capabilities, it is therefore necessary to use RS232 communications to achieve basic PID operation.

d) As 25 data register are required for the setup parameters for the PID loop, the head address of this data stack cannot be greater than D975. The contents of this data stack are explained later in this section. Multiple PID instructions can be programmed, however each PID loop must not have conflicting data registers.

e) There are control limits in the PLC intended to help the PID controlled machines operate in a safe manner. If it becomes necessary to reset the Set Point Value (S_1) during operation, it is recommended to turn the PID command Off and restore the command after entering the new Set Point Value. This will prevent the safety control limits from stopping the operation of the PID instruction prematurely.

f) The PID instruction has a special set of error codes associated with it. Errors are identified in the normal manner. The error codes associated with the PID loop will be flagged by M8067 with the appropriate error code being stored in D8067. These error devices are not exclusive to the PID instruction so care should be taken to investigate errors properly. Please see chapter 6, ‘Diagnostic Devices’ for more information.

g) A full PID iteration does not have to be performed. By manipulation of the setup parameters P (proportional), I (Integral) or D (derivative) loops may be accessed individually or in a user defined/selected group. This is detailed later in this section.

PID Equations

Forward $PV_{nf} > SV$

$$\Delta MV = K_P\left\{(EV_n - EV_{(n-1)}) + \frac{T_S}{T_I}EV_n + D_n\right\}$$

$$EV_n = PV_{nf} - SV$$

$$D_n = \frac{T_D}{T_S + K_D \cdot T_D}(-2PV_{nf-1} + PV_{nf} + PV_{nf-2}) + \frac{K_D \cdot T_D}{T_S + K_D \cdot T_D} \cdot D_{n-1}$$

$$MV_n = \sum \Delta MV$$

STOP

$SV > PV_{nf}$

$$\Delta MV = K_P\left\{(EV_n - EV_{n-1}) + \frac{T_S}{T_I}EV_n + D_n\right\}$$

$$EV_n = SV - PV_{nf}$$

$$D_n = \frac{T_D}{T_S + K_D \cdot T_D}(2PV_{nf-1} - PV_{nf} - PV_{nf-2}) + \frac{K_D \cdot T_D}{T_S + K_D \cdot T_D} \cdot D_{n-1}$$

$$MV_n = \sum \Delta MV\Delta$$

$PV_{nf} = PV_n + \alpha(PV_{nf-1} - PV_n)$

EV_n = the current Error Value
EV_{n-1} = the previous Error Value
SV = the Set Point Value (S_1)
PV_n = the current Process Value (S_2)
PV_{nf} = the calculated Process Value
PV_{nf-1} = the previous Process Value
PV_{nf-2} = the second previous Process Value
ΔMV = the change in the Output Manipulation Values
MV_n = the current Output Manipulation Value (D)
D_n = the Derivative Value
D_{n-1} = the previous Derivative Value
K_P = the Proportion Constant
α = the Input Filter
T_S = the Sampling Time
T_I = the Integral Time Constant
T_D = the Time Derivative Constant
K_D = the Derivative Filter Constant

Please see the Parameter setup section for a more detailed description of the variable parameters and in which memory register they must be set.

Forward and Reverse operation (S_3+1, b0)

The Forward operation is the condition where the Process Value, PV_{nf}, is greater than the Set Point, SV. An example is a building that requires air conditioning. Without air conditioning, the temperature of the room will be higher than the Set Point so work is required to lower PV_{nf}.

The Reverse operation is the condition where the Set Point is higher than the Process Value. An example of this is an oven. The temperature of the oven will be too low unless some work is done to raise it, i.e. - the heating element is turned On.

The assumption is made with PID control that some work will need to be performed to bring the system into balance. Therefore, ΔMV will always have a value. Ideally, a system that is stable will require a constant amount of work to keep the Set Point and Process Value equal.

PID setup parameters; S3

The PID setup parameters are contained in a 25 register data stack. Some of these devices require data input from the user, some are reserved for the internal operation and some return output data from the PID operation.

Parameters S_3+0 through S_3+6 must be set by the user.

Parameter S3 + P	Parameter name/function	Description		Setting range
S3+0	Sampling time **TS**	The time interval set between the reading the current Process Value of the system (PV_{nf})		1 to 32767 msec
S3+1	Action - reaction direction and alarm control	b0	Forward operation(0), Reverse operation (1)	Not applicable
		b1	Process Value (PV_{nf}) alarm enable, OFF(0)/ ON(1)	
		b2	Output Value (MV) alarm enable, OFF(0)/ON(1)	
		b3 - 15	Reserved	
S3+2	Input filter α	Alters the effect of the input filter.		0 to 99%
S3+3	Proportional gain **KP**	This is a factor used to align the proportional output in a known magnitude to the change in the Process Value (PV_{nf}). This is the **P** part of the PID loop.		1 to 32767%
S3+4	Integral time constant **TI**	This is the **I** part of the PID loop. This is the time taken for the corrective integral value to reach a magnitude equal to that applied by the proportional or **P** part of the loop. Selecting 0 (zero) for this parameter disables the **I** effect.		(0 to 32767) x 100 msec
S3+5	Derivative gain **KD**	This is a factor used to align the derivative output in a known proportion to the change in the Process Value (PV_{nf})		1 to 100%
S3+6	Derivative time constant **TD**	This is the **D** part of the PID loop. This is the time taken for the corrective derivative value to reach a magnitude equal to that applied by the proportional or **P** part of the loop. Selecting 0 (zero) for this parameter disables the **D** effect.		(0 to 32767) x 10 msec
S3+7 to S3+19	Reserved for use for the internal processing			
S3+20	Process Value, maximum positive change	Active when S3+1, b1 is set ON.	This is a user defined maximum limit for the Process Value (PVnf). If the Process Value (PVnf) exceeds the limit, S3+24, bit b0 is set On.	0 to 32767
S3+21	Process Value, minimum value		This is a user defined lower limit for the Process Value. If the Process Value (PVnf) falls below the limit, S3+24, bit b1 is set On.	
S3+22	Output Value, maximum positive change	Active when S3+1, b2 is set ON.	This is a user defined maximum limit for the quantity of positive change which can occur in one PID scan. If the Output Value (MV) exceeds this, S3+24, bit b2 is set On.	
S3+23	Output Value, maximum negative change		This is a user defined maximum limit for the quantity of negative change which can occur in one PID scan. If the Output Value (MV) falls below the lower limit, S3+24, bit b3 is set On.	
S3+24	Alarm flags **(Read Only)**	b0	High limit exceeded in Process Value (PVnf)	Not applicable
		b1	Below low limit for the Process Value (PVnf)	
		b2	Excessive positive change in Output Value (MV)	
		b3	Excessive negative change in Output Value (MV)	
		b4 - 15	Reserved	

See **Initial values for PID loops** for basic guidance on initial PID values; page 5-114.
See page 10-24 for additional parameters available with FX2N, FX2NC & FX1N MPUs.

Configuring the PID loop

The PID loop can be configured to offer variations on PID control. These are as follows:

Control method	Selection via setup registers			Description
	S_3 +3 (**K_P**)	S_3+ 4 (**T_I**)	S_3 + 6 (**T_D**)	
P	User value	Set to 0 (zero)	Set to 0 (zero)	Proportional effect only
PI	User value	User value	Set to 0 (zero)	Proportional and integral effect
PD	User value	Set to 0 (zero)	User value	Proportional and derivative effect
PID	User value	User value	User value	Full PID

It should be noted that in all situations there must be a proportional or 'P' element to the loop.

P - proportional change

When a proportional factor is applied, it calculates the difference between the Current Error Value, EV_n, and the Previous Error Value, EV_{n-1}. The Proportional Change is based upon how fast the Process Value is moving closer to (or further away from) the Set Point Value NOT upon the actual difference between the PV_{nf} and SV.

Note: Other PID systems might operate using an equation that calculates the Proportional change based upon the size of the Current Error Value only.

I - integral change

Once a proportional change has been applied to an error situation, 'fine tuning' the correction can be performed with the I or integral element.
Initially only a small change is applied but as time increases and the error is not corrected the integral effect is increased. It is important to note how T_I actually effects how fast the total integral correction is applied. The smaller T_I is, the bigger effect the integral will have.

Note: The T_I value is set in data register S_3+4. Setting zero for this variable disables the Integral effect.

The Derivative Change

The derivative function supplements the effects caused by the proportional response. The derivative effect is the result of a calculation involving elements T_D, T_S, and the calculated error. This causes the derivative to initially output a large corrective action which dissipates rapidly over time. The speed of this dissipation can be controlled by the value T_D: If the value of T_D is small then the effect of applying derivative control is increased.

Because the initial effect of the derivative can be quite severe there is a 'softening' effect which can be applied through the use of **K_D**, the derivative gain. The action of **K_D** could be considered as a filter allowing the derivative response to be scaled between 0 and 100%.

The phenomenon of chasing, or overcorrecting both too high and too low, is most often associated with the Derivative portion of the equation because of the large initial correction factor.

Note: The T_D value is set in Data register S_3+6. Setting zero for this variable disables the Derivative effect.

Effective use of the input filter α S_3+2

To prevent the PID instruction from reacting immediately and wildly to any errors on the Current Value, there is a filtering mechanism which allows the PID instruction to observe and account for any significant fluctuations over three samples.
The quantitative effect of the input filter is to calculate a filtered Input Value to the PID instruction taken from a defined percentage of the Current Value and the previous two filtered Input Values.
This type of filtering is often called first-order lag filter. It is particularly useful for removing the effects of high frequency noise which may appear on input signals received from sensors.
The greater the filter percentage is set the longer the lag time. When the input filter is set to zero, this effectively removes all filtering and allows the Current Value to be used directly as the Input Value.

Initial values for PID loops

The PID instruction has many parameters which can be set and configured to the user's needs. The difficulty is to find a good point from which to start the fine tuning of the PID loop to the system requirements. The following suggestions will not be ideal for all situations and applications but will at least give users of the PID instruction a reasonable points from which to start.
A value should be given to all the variables listed below before turning the PID instruction ON. Values should be chosen so that the Output Manipulated Value does not exceed ± 32767.
Recommended initial settings:

T_S = Should be equal to the total program scan time or a multiple of that scan time, i.e. 2 times, 5 times, etc.
α = 50%
K_P = This should be adjusted to a value dependent upon the maximum corrective action to reach the set point - values should be experimented with from an arbitrary 75%
T_I = This should ideally be 4 to 10 times greater than the T_D time
K_D = 50%
T_D = This is set dependent upon the total system response, i.e. not only how fast the programmable controller reacts but also any valves, pumps or motors.
For a fast system reaction T_D will be set to a quick or small time, this should however never be less than T_S. A slower reacting system will require the T_D duration to be longer. A beginning value can be T_D twice the value of T_S.

Care should be taken when adjusting PID variables to ensure the safety of the operator and avoid damage to the equipment.

On FX2N MPUs pre-tuning feature is available that can quickly provide initial values for the PID process. Refer to page 10-28 for more details.

With ALL PID values there is a degree of experimentation required to tune the PID loop to the exact local conditions. A sensible approach to this is to adjust one parameter at a time by fixed percentages, i.e. say increasing (or decreasing) the K_P value in steps of 10%. Selecting PID parameters without due consideration will result in a badly configured system which does not perform as required and will cause the user to become frustrated. Please remember the PID process is a purely mathematical calculation and as such has no regard for the 'quality' of the variable data supplied by the user/system - the PID will always process its PID mathematical function with the data available.

Example PID Settings

The partial program shown at below demonstrates which parameters must be set for the functioning of the FX2N. The first step sets the user values for S_3+0 to S_3+6. The PID instruction will be activated when M4 is On.

From the PID instruction at the bottom of the ladder, S_1 = D200; S_2 = D201; S_3 = D500; and D or MV = D525.

D500: Ts = 500 ms

D501: Forward Operation, Alarms Not Enabled

D502: Input Filter = 50%

D503: K_P = 75%

D504: T_I = 4000 ms

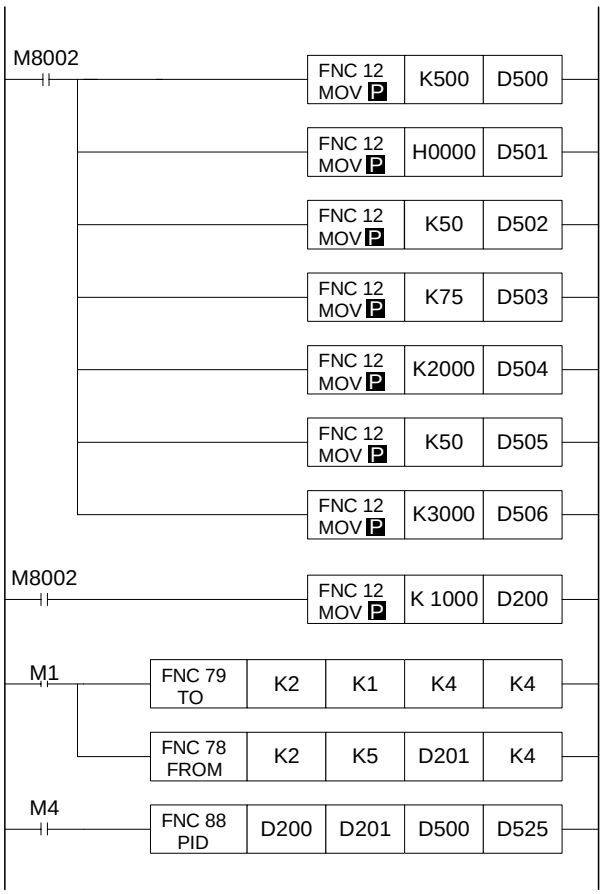
D505: K_D = 50%

D506: T_D = 1000 ms

D200: Set Point = 1000

D201: PV_{nf} (an analog input value)

Begin the PID instruction
D525: PID Output Value



MEMO

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.10 Floating Point 1 & 2 - Functions 110 to 129

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

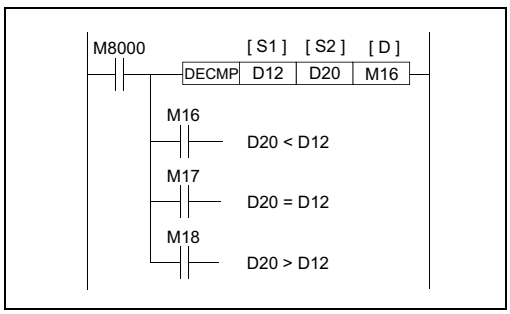
✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.10.1 ECMP (FNC 110)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
ECMP FNC 110 (Floating Point Compare)	Compares two floating point values - results of <, = and > are given	K, H - integer value automatically converted to floating point D - must be in floating point format (32bits).		Y, M, S Note: 3 consecutive devices are used.	DECMP, DECMPP: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The data of S_1 is compared to the data of S_2. The result is indicated by 3 bit devices specified with the head address entered as D. The bit devices indicate:

S_2 is less than < S_1 - bit device D is ON
S_2 is equal to = S_1 - bit device D_{+1} is ON
S_2 is greater than > S_1 - bit device D_{+2} is ON

Points to note:

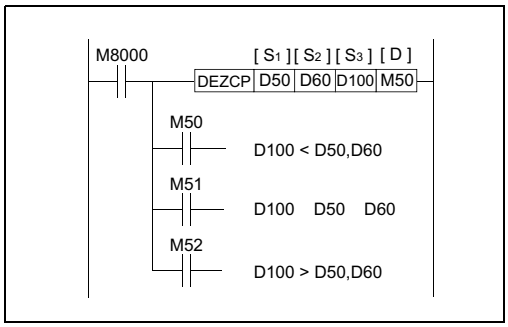
The status of the destination devices will be kept even if the ECMP instruction is deactivated.
Full algebraic comparisons are used: i.e. -1.79×10^{27} is smaller than 9.43×10^{-15}

5.10.2 EZCP (FNC 111)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S_1	S_2	S_3	D	
EZCP FNC 111 (Floating Point Zone Compare)	Compares a float range with a float value - results of <, = and > are given	K, H - integer value automatically converted to floating point D - must be in floating point format (32 bits). **Note:** S_1 must be less than S_2			Y, M, S Note: 3 consecutive devices are used.	DEZCP, DEZCPP: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

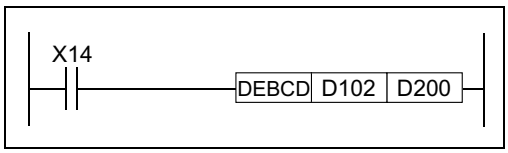
The operation is the same as the ECMP instruction except that a single data value (S_3) is compared to a data range (S_1 - S_2).
S_3 is less than S_1 and S_2 - bit device D is ON
S_3 is between S_1 and S_2 - bit device D_{+1} is ON
S_3 is greater than S_2 - bit device D_{+2} is ON

5.10.3 EBCD (FNC 118)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
EBCD FNC 118 (Float to Scientific conversion)	Converts floating point number format to scientific number format	D - must be in floating point format (32 bits).	D - 2 consecutive devices are used D - mantissa D+1 - exponent.	DEBCD, DEBCDP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

Converts a floating point value at S into separate mantissa and exponent parts at D and D+1 (scientific format).

Points to note:

a) The instruction must be double word format. The destinations D and D+1 represent the mantissa and exponent of the floating point number respectively.

b) To provide maximum accuracy in the conversion the mantissa D will be in the range 1000 to 9999 (or 0) and the exponent D+1 corrected to an appropriate value.

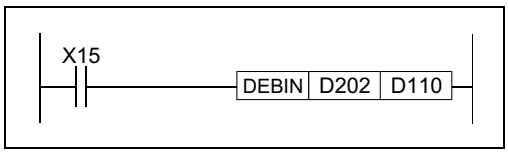
c) E.g. S= 3.4567×10^{-5} will become D= 3456, D+1 = -8

5.10.4 EBIN (FNC 119)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
EBIN FNC 119 (Scientific to Float conversion)	Converts scientific number format to floating point number format	D - 2 consecutive devices are used S- mantissa S+1 - exponent.	D - a floating point value (32 bits).	DEBIN, DEBINP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

Generates a floating point number at D from scientific format data at source S.

Points to note:

a) The instruction must be double word format. The source data Sand S+1 represent the mantissa and exponent of the floating point number to be generated.

b) To provide maximum accuracy in the conversion the mantissa S must be in the range 1000 to 9999 (or 0) and the exponent S+1 corrected to an appropriate value.

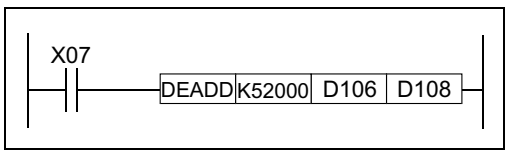
c) E.g. S= 5432, S+1 = 12 will become D= 5.432 x 10^{9}

5.10.5 EADD (FNC 120)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
EADD FNC 120 (Floating Point Addition)	Adds two floating point numbers together	K, H - integer value automatically converted to floating point D - must be in floating point format (32 bits).		D - a floating point value (32 bits).	DEADD, DEADDP: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Zero M8020 Borrow M8021 Carry M8022



Operation:

The floating point values stored in the source devices S_1 and S_2 are algebraically added and the result stored in the destination device D.

Points to note:

a) The instruction must use the double word format; i.e., **D**EADD or **D**EADDP. All source data and destination data will be double word; i.e. uses two consecutive data registers to store the data (32 bits).
Except for K or H, all source data will be regarded as being in floating point format and the result stored in the destination will also be in floating point format.

b) If a constant K or H is used as source data, the value is converted to floating point before the addition operation.

c) The addition is mathematically correct: i.e., $2.3456 \times 10^{2} + (-5.6 \times 10^{-1}) = 2.34 \times 10^{2}$

d) The same device may be used as a source and as the destination. If this is the case then, on continuous operation of the DEADD instruction, the result of the previous operation will be used as a new source value and a new result calculated.
This will happen every program scan unless the pulse modifier or an interlock program is used.

e) If the result of the calculation is zero “0” then the zero flag, M8020 is set ON.
If the result of the calculation is larger than the largest floating point number then the carry flag, M8021 is set ON and the result is set to the largest value.
If the result of the calculation is smaller than the smallest floating point number then the borrow flag, M8022 is set ON and the result is set to the smallest value.

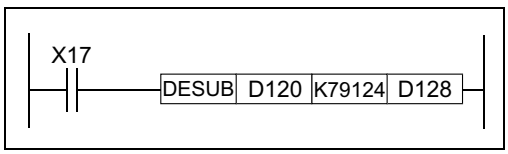
For more information about the format of floating point number refer to page 4-46.

5.10.6 EAUB (FNC 121)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
ESUB FNC 121 (Floating Point Sub-traction)	Subtracts one floating point number from another	K, H - integer value automatically converted to floating point D - must be in floating point number format (32 bits).		D - a floating point value (32 bits).	DESUB, DESUBP: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Zero M8020 Borrow M8021 Carry M8022



Operation:

The floating point value of S_2 is subtracted from the floating point value of $S_1$and the result stored in destination device D.

Points to note:

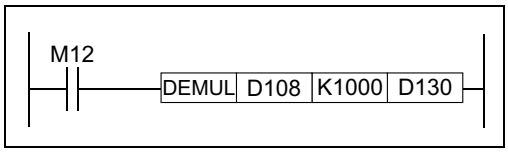
All points of the EADD instruction apply, except that a subtraction is performed.
See page 5-122.

5.10.7 EMUL (FNC 122)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
EMUL FNC 122 (Floating Point Multiplication)	Multiplies two floating point numbers together	K, H - integer value automatically converted to floating point D - must be in floating point format (32 bits).		D - a floating point value (32 bits).	DEMUL, DEMULP: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The floating point value of S_1 is multiplied with the floating point value of S_2. The result of the multiplication is stored at D as a floating point value.

Points to note:

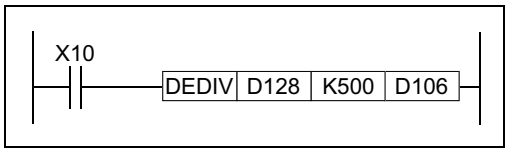
Point a, b, c and d of the EADD instruction apply, except that a multiplication is performed.
See page 5-122.

5.10.8 EDIV (FNC 123)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
EDIV FNC 123 (Floating Point Division)	Divides one floating point number by another.	K, H - integer value automatically converted to floating point D - must be in floating point format (32 bits).		D - a floating point value (32 bits).	DEDIV, DEDIVP: 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The floating point value of S1 is divided by the floating point value of S2. The result of the division is stored in D as a floating point value. No remainder is calculated.

Points to note:

Points a, b, c, d of the EADD instruction apply, except that a division is performed.
See page 5-122.

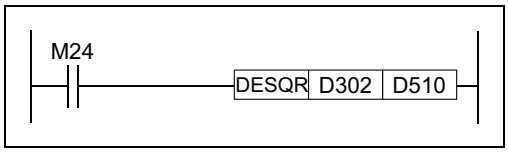
- If S2 is 0 (zero) then a divide by zero error occurs and the operation fails.

5.10.9 ESQR (FNC 127)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
ESQR FNC 127 (Floating Point Square Root)	Calculates the square root of a floating point value.	K, H - integer value automatically converted to floating point D - must be in floating point number format (32 bits).	D - a floating point value (32 bits).	DESQR, DESQRP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Zero M8020



Operation:

A square root is performed on the floating point value of Sand the result is stored in D.

Points to note:

Points a, b, c, d of the EADD instruction apply, except that a square root is performed.
See page 5-122.

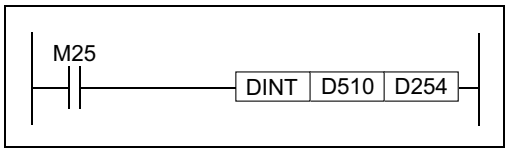
- If S is negative then an error occurs and error flag M8067 is set ON.

5.10.10 INT (FNC 129)

FX1S | FX1N | FX2N | FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
INT FNC 129 (Float to Integer)	Converts a number from floating point format to decimal format	D - must be in floating point number format (always 32 bits).	D - decimal format for INT, INTP - 16 bits for DINT, DINTP - 32 bits	INT, INTP: 5 steps DINT, DINTP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Zero M8020 Borrow M8021 Carry M8022



Operation:

The floating point value of S is rounded down to the nearest integer value and stored in normal binary format in D.

Points to note:

a) The source data is always a double (32 bit) word; a floating point value.
For single word (16 bit) operation the destination is a 16 bit value.
For double word (32 bit) operation the destination is a 32 bit value.

b) This instruction is the inverse of the FLT instruction. (See page 5-49)

c) If the result is 0 then the zero flag M8020 is set ON.
If the source data is not a whole number it must be rounded down. In this case the borrow flag M8021 is set ON to indicate a rounded value.
If the resulting integer value is outside the valid range for the destination device then an overflow occurs. In this case the carry flag M8022 is set on to indicate overflow.
Note: If overflow occurs, the value in the destination device will not be valid.

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.11 Trigonometry - FNC 130 to FNC 139

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

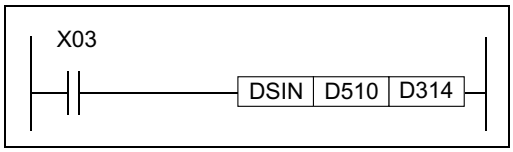
⌧ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.11.1 SIN (FNC 130)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
SIN FNC 130 (Sine)	Calculates the sine of a floating point value	D - must be in floating point number format (32 bits).(radians)	D - a floating point value (32 bits).	DSIN, DSINP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Contents:

This instruction performs the mathematical SIN operation on the floating point value in S. The result is stored in D.

Points to note:

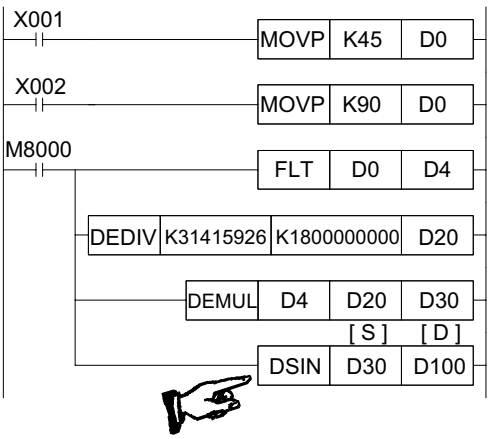
a) The instruction must use the double word format: i.e., **D**SIN or **D**SINP. All source and destination data will be double word; i.e., uses two consecutive data registers to store the data (32 bits).
The source data is regarded as being in floating point format and the destination is also in floating point format.

b) The source value must be an angle between 0 to 360 degrees in radians; i.e.,

$$0° \leq S < 360°$$

Radian Angles

Below is an program example of how to calculate angles in radians using floating point.



K45 degrees to D0

K90 degrees to D0

Convert D0 to float in D4,D5

Calculate π in radians (π/180)
Store as a float in D20,D21

Calculate angle in radians in D30,D31
(deg° × π/180 = rads)

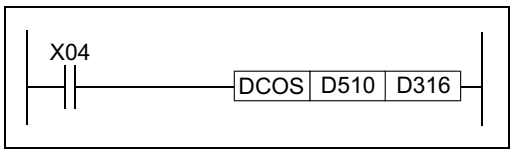
Calculate SIN of angle in D100

5.11.2 COS (FNC 131)

FX1S | FX1N | FX2N | FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
COS FNC 131 (Cosine)	Calculates the cosine of a floating point value	D - must be in floating point number format (32 bits).	D - a floating point value (32 bits).	DCOS, DCOSP: 9 steps

16 BIT OPERATION | 32 BIT OPERATION | PULSE-P



Contents:

This instruction performs the mathematical COS operation on the floating point value in S. The result is stored in D.

Points to note:

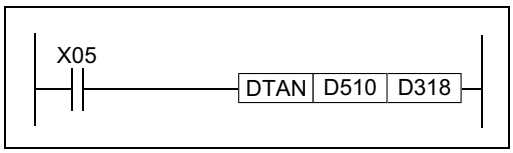
All the points for the SIN instruction apply, except that COS is calculated.
See page 5-127.

5.11.3 TAN (FNC 132)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
TAN FNC132 (Tangent)	Calculates the tangent of a floating point value	D - must be in floating point number format (32 bits).	D - a floating point value (32 bits).	DTAN, DTANP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Contents:

This instruction performs the mathematical TAN operation on the floating point value in S. The result is stored in D.

Points to note:

All the points for the SIN instruction apply, except that COS is calculated.
See page 5-127.

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.12 Data Operations 2 - FNC 140 to FNC 149

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

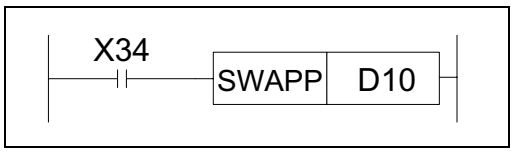
✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.12.1 SWAP (FNC 147)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		S	
SWAP FNC 147 (Byte Swap) ✈	The high and low byte of the designated devices are exchanged	KnY, KnM, KnS, T, C, D, V, Z	SWAP,SWAPP : 5 steps DSWAP, DSWAPP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Contents:

The upper byte and the lower byte of the source device are swapped.
This instruction is equivalent to operation 2 of FNC 17 XCH (see page 5-21).

Points to note:

a) In single word (16 bit) operation the upper and lower byte of the source device are exchanged.

b) In double word (32 bit) operation the upper and lower byte of each or the two 16 bit devices are exchanged.
Result of DSWAP(P) D10:

Values are in Hex for clarity		Before DSWAP	After DSWAP
D10	Byte 1	1F$_H$	8B$_H$
	Byte 2	8B$_H$	1F$_H$
D11	Byte 1	C4$_H$	35$_H$
	Byte 2	35$_H$	C4$_H$

c) If the operation of this instruction is allowed to execute each scan, then the value of the source device will swap back to its original value every other scan. The use of the pulse modifier or an interlock program is recommended.

MEMO

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.13 Positioning Control - FNC 150 to FNC 159

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.13.1 Cautions when using Positioning Instructions

FX1S	FX1N	FX2N	FX2NC

The following positioning instructions are application instructions that can be used many times in a program.
When designing a program, make sure to follow the cautions outlined below with regard to instruction drive timing.

FNC 156 (ZRN)
FNC 157 (PLSV)
FNC 158 (DRVI)
FNC 159 (DRVA)

- Do not drive positioning instructions which use the same output relay (Y000 or Y001) at the same time. If such instructions are driven at the same time, they will be treated as double coils, and not function correctly.
- Before setting a drive contact ON after is has been set to OFF be sure that the following condition is satisfied;
  One or more operation cycles of the 'pulse output monitor (Y000:M8147, Y001:M8148)' must occour after the positioning instruction is turned OFF, before it can be used again.
  This condition must be met, as one or more OFF operations are required for the re-driving of a positioning instruction.
  If it is not met, and 'operation error' will occour during the instruction execution.
- Use the Step Ladder Program to correctly set up positioning instructions in conformance to the cautions above.

Caution when using Positioning instructions with FNC 57 (PLSY) & FNC 59 (PLSR)

- Pulse output instructions FNC 57 (PLSY) & FNC 59 (PLSR) use output points Y000 and Y001 in the same way as the positioning instructions described above.
- If a positioning and a pulse output instruction are used in the same operation, the conflicting instructions will be treated as double coils and not function correctly.
- It is recommended to use a FNC 158 (DRVI) instruction in place of either a FNC 57 (PLSY) or FNC 59 (PLSR) instruction to avoid incorrect operation when pulse outputs are required while positioning instructions are being used.

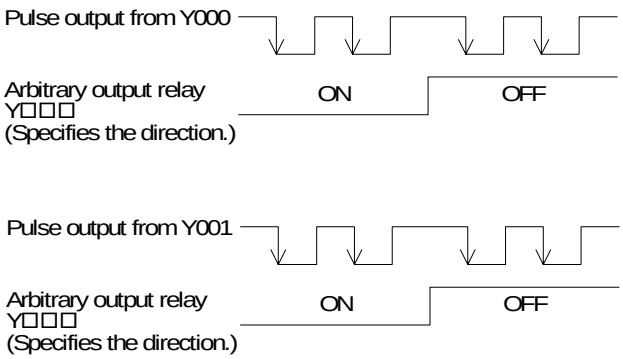
Output terminals Y000 and Y001 are high speed response type

Voltage range : 5 to 24V DC
Current range : 10 to 100mA
Output frequency : 100kHz or less

5.13.2 Pulse train settings

FX1S	FX1N	FX2N	FX2NC

When a positioning operation is executed from the PLC, the pulse output signal has the 'Pulse train + Sign' format during control, as shown in the figure below.



Make sure to set the pulse train input mode on the servo amplifier or stepper motor as follows;

Pulse train input mode: Pulse train + Sign
Pulse train logic: negative logic

5.13.3 Devices related to positioning

FX1S	FX1N	FX2N	FX2NC

Device No.		Data size	Initial value	Description
D8140	Lower	32 bit	0	Operates as current value registers of positioning instruction output to Y000 For FNC 157 (PLSV), FNC 158 (DRVI), FNC 159 (DRVA) instructions, current value increases or decreases in accordance with direction of rotation. Although FNC 57 (PLSY) and FNC 59 (PLSR) instructions use the same current value registers, the current value represents the accumulating total number of output pulses during instruction execution.
D8141	Upper			
D8142	Lower	32 bit	0	Operates as current value registers of positioning instruction output to Y001 For FNC 157 (PLSV), FNC 158 (DRVI), FNC 159 (DRVA) instructions, current value increases or decreases in accordance with direction of rotation. Although FNC 57 (PLSY) and FNC 59 (PLSR) instructions use the same current value registers, the current value represents the accumulating total number of output pulses during instruction execution.
D8143	Upper			
D8145		16 bit	0	Bias speed when FNC 156 (ZRN), FNC 158 (DRVI), FNC 159 (DRVA) instructions are executed Set range: 1/10 or less of maximum speed (D8146 & D8147) If the current value exceeds this range, it is automatically set to 1/10 of the maximum speed during operation.
D8146	Lower	32 bit	100,000	Maximum speed when FNC 156 (ZRN), FNC 158 (DRVI), FNC 159 (DRVA) instructions are executed. Set range 10 to 100,000 (Hz).
D8147	Upper			
D8148		16 bit	100	Acceleration/Deceleration time in which maximum speed (D8146 & D8147) is achieved from bias speed (D8145) when FNC 156 (ZRN), FNC 158 (DRVI), FNC 159 (DRVA) instructions are executed. Set range 50 to 5,000 (ms)

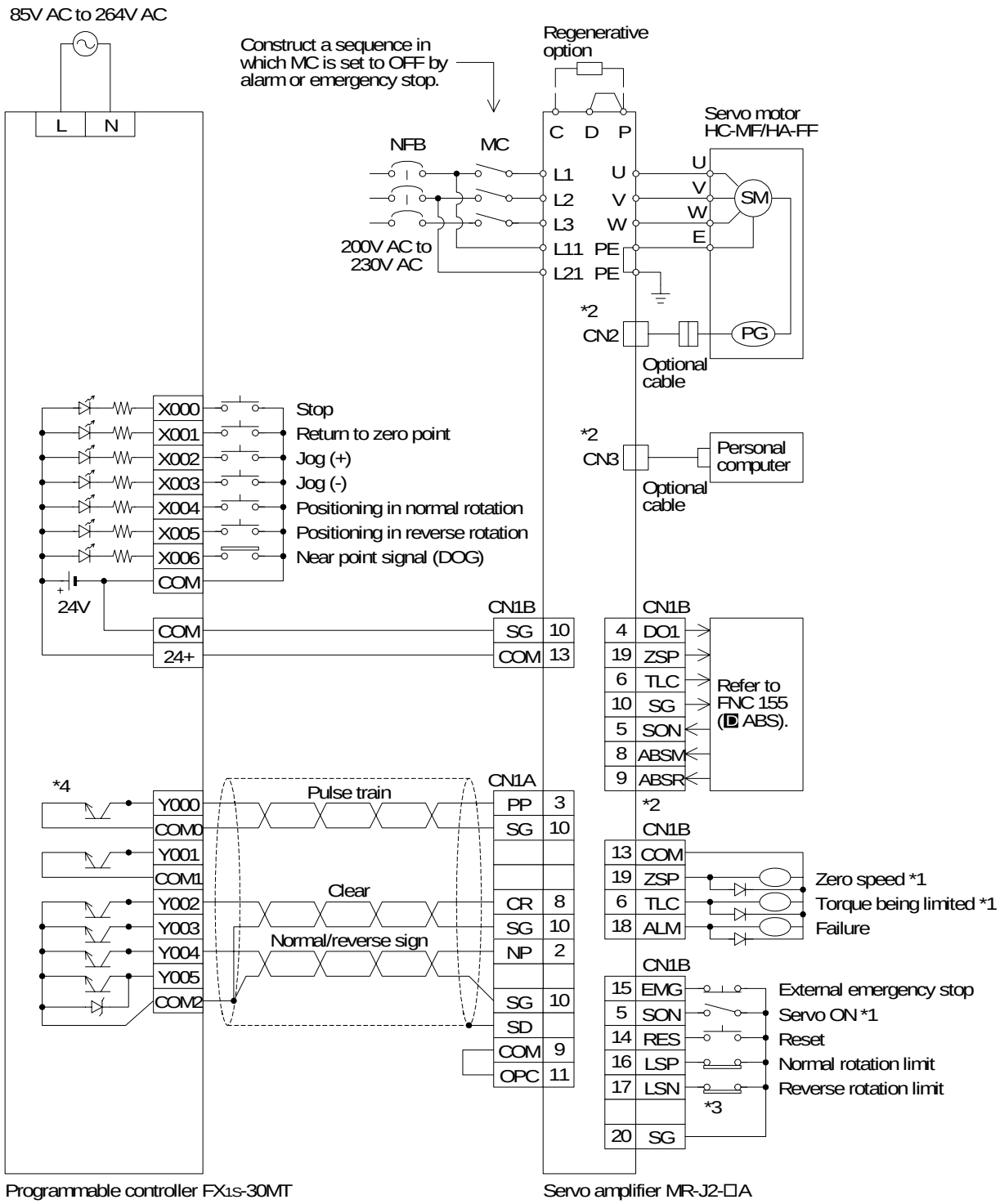
Device No.	Attribute	Description
M8145	Drive enable	Y000 pulse output stop command (immediate stop)
M8146	Drive enable	Y001 pulse output stop command (immediate stop)
M8147	Read only flag	Y000 pulse output monitor (BUSY/READY)
M8148	Read only flag	Y001 pulse output monitor (BUSY/READY)

5.13.4 Servo Wiring Example

FX1S	FX1N	FX2N	FX2NC

Example of connection to a Mitsubishi MR-J2-*A servo.

Note. The PLC required for this connection is a SINK Transistor output type.

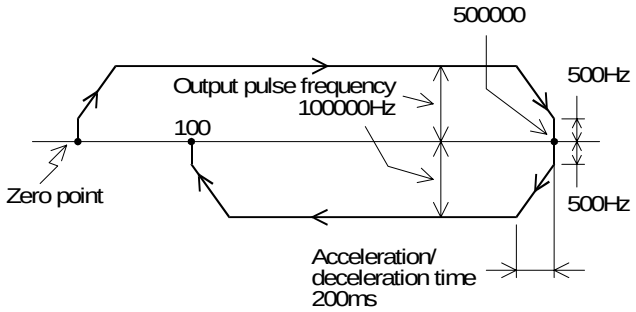


*1 Connect to programmable controller when absolute position detection is required.
*2 Ports CN1A, CN1B, CN2 & CN3 are the same shape. Do not confuse them.
*3 Connect a limit switch to the servo amplifier.
*4 ONLY use a transistor output type PLC.

5.13.5 Example Program

FX1S	FX1N	FX2N	FX2NC

The following example program for forward/reverse operation uses the I/O assignment shown in section 5.13.4 Servo Wiring Example.
During operation positioning is performed using the absolute position method shown below.



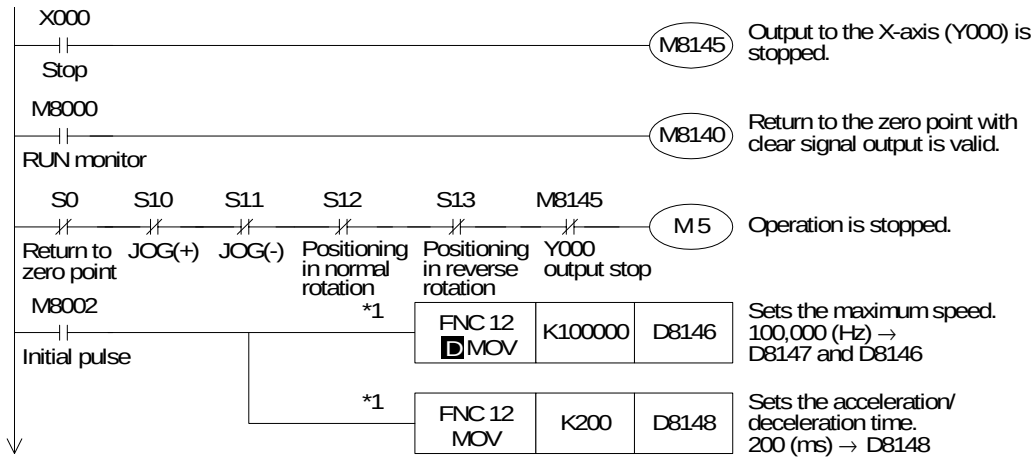
In this example the actual output frequency for the first step, acceleration, and the last step, deceleration, can be obtained using the following expression.

$$\sqrt{A \div (2 \times (C \div 1000))} = \text{Output}$$

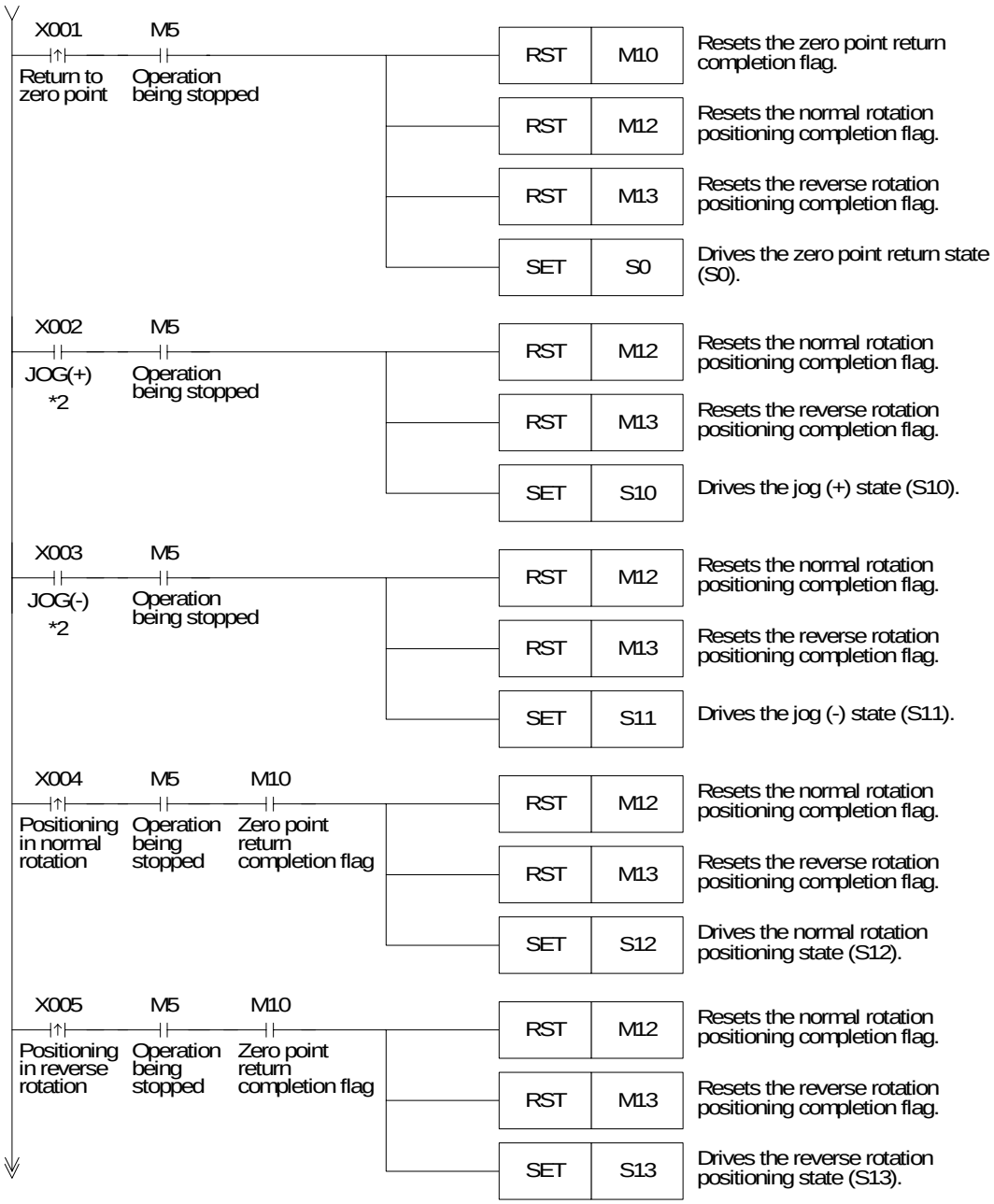
$$\sqrt{100000 \div (2 \times (200 \div 1000))} = 500\text{Hz}$$

A = Maximum speed (D8146, D8147)
B = Acceleration/Deceration time

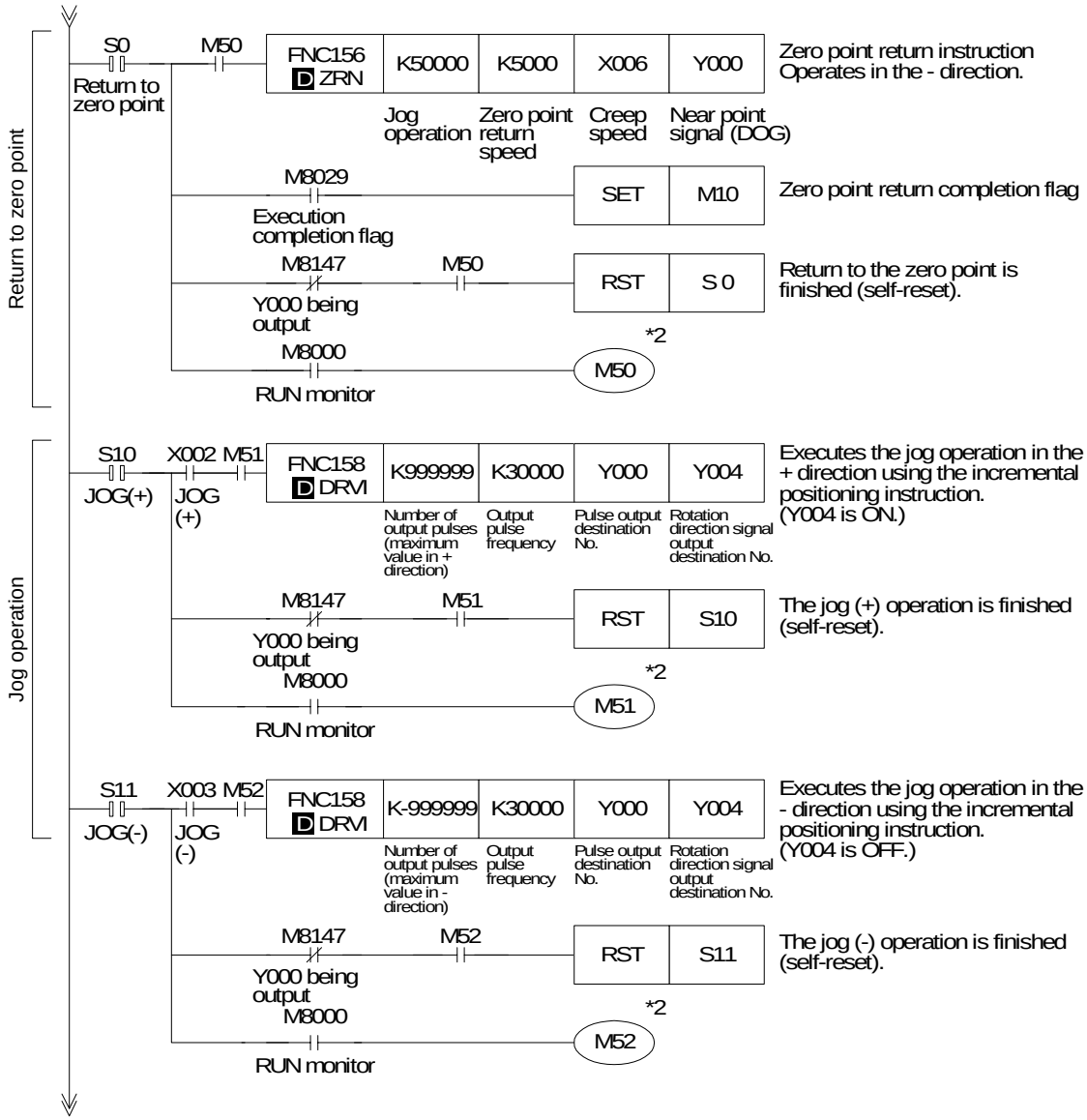
Step Ladder program.

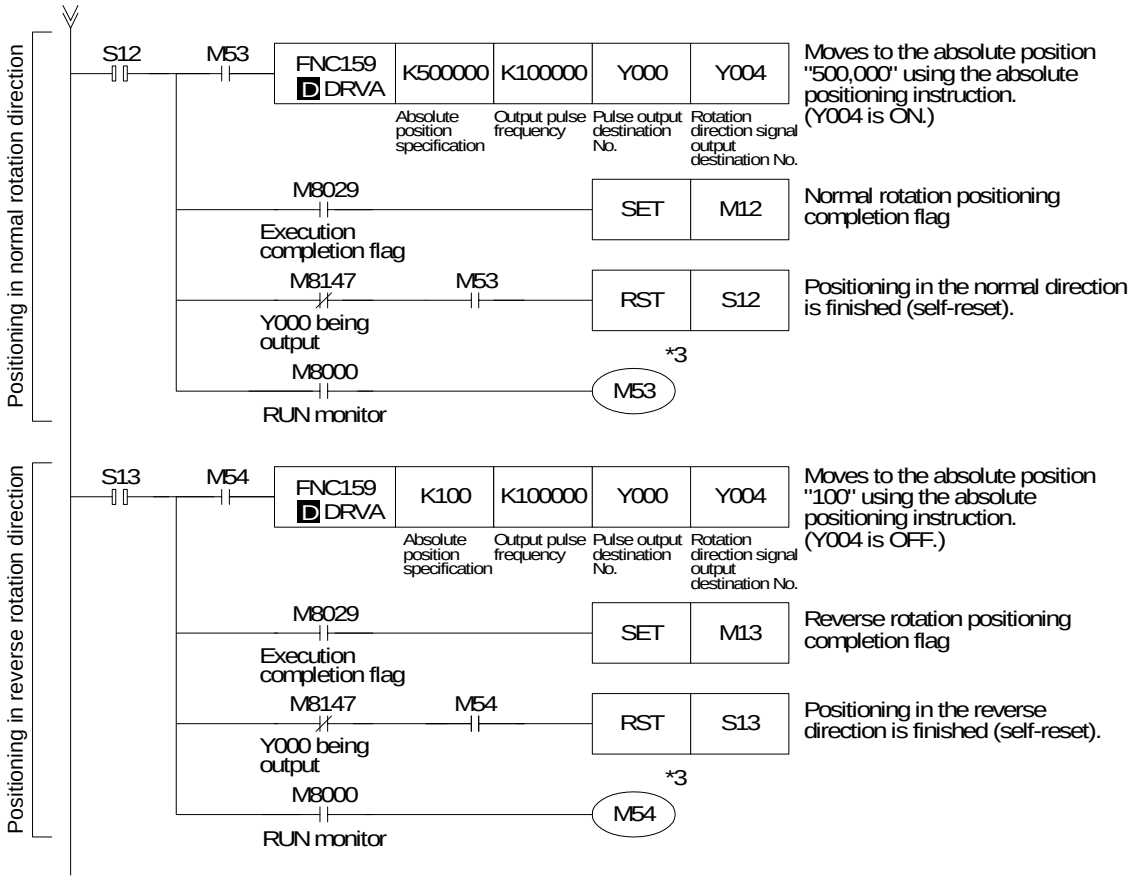


*1 When the maximum speed or Acceleration/deceleration do not have to be changed from their initial values, programming is not required.



*2 The maximum size of a JOG command is 999,999 pulses, as this is the maximum number of output pulses for a FNC 158 (DRVI) instruction. If a greater distance is required execute more than one JOG command.


Return to zero point
S0
M50
Return to zero point
FNC156
D ZRN
K50000
K5000
X006
Y000
Zero point return instruction
Operates in the - direction.
Jog operation
Zero point return speed
Creep speed
Near point signal (DOG)
M8029
Execution completion flag
SET
M10
Zero point return completion flag
M8147
Y000 being output
M50
RST
S 0
Return to the zero point is finished (self-reset).
*2
M8000
RUN monitor
M50
Jog operation
S10
X002
M51
JOG(+)
JOG (+)
FNC158
D DRVI
K999999
K30000
Y000
Y004
Executes the jog operation in the + direction using the incremental positioning instruction.
(Y004 is ON.)
Number of output pulses (maximum value in + direction)
Output pulse frequency
Pulse output destination No.
Rotation direction signal output destination No.
M8147
Y000 being output
M51
RST
S10
The jog (+) operation is finished (self-reset).
*2
M8000
RUN monitor
M51
S11
X003
M52
JOG(-)
JOG (-)
FNC158
D DRVI
K-999999
K30000
Y000
Y004
Executes the jog operation in the - direction using the incremental positioning instruction.
(Y004 is OFF.)
Number of output pulses (maximum value in - direction)
Output pulse frequency
Pulse output destination No.
Rotation direction signal output destination No.
M8147
Y000 being output
M52
RST
S11
The jog (-) operation is finished (self-reset).
*2
M8000
RUN monitor
M52



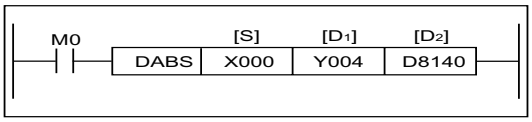
*3 The instruction drive timing is delayed by one opeation cycle to prevent simultaneous driving of positioning instructions.

5.13.6 ABS (FNC 155)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D_1	D_2	
ABS FNC 155 Absolute current value read	Reads the absolute position from a servo motor	X,Y,M,S	Y,M,S	T,C,D,V,Z	DABS 13 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

This instruction reads the absolute position data when a Mitsubishi servo motor, MR-H or MR-J2, equipped with absolute positioning function is connected. [S] is the first of three inputs used for communication flags (see drawing below), [D_1] is the first of three communication outputs and [D_2] is the data destination register.

Points to note:

a)This instruction is 32-Bit. Be sure to input as “DABS”

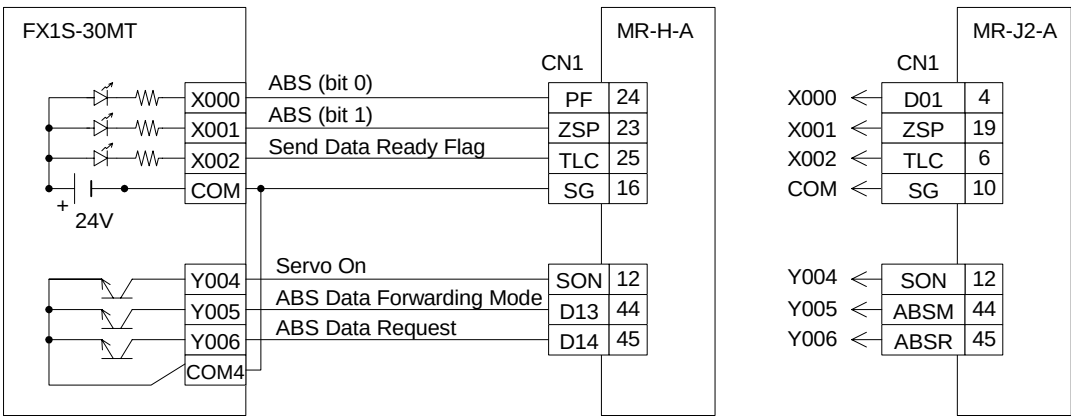
b)Read starts when the instruction drive contact turns ON. When the read is complete, the execution complete flag M8029 is energized.
If the instruction drive contact is turned OFF during read, read is aborted.

c)When designing a system, set the servo amplifier to be ON earlier than the power of the PLC, or so that they are both powered ON at the same time.

d)The device [D_2] to which the absolute value is read, can be set within a word device range. However, the absolute value should be transferred at some point to the correct registers (D8141 & D8140)

e)The DABS instruction drive contact uses an input which is always ON, even after the absolute value is read.
If the instruction drive contact turns OFF after the read is complete, the servo ON (SON) signal will turn OFF and the operation disabled.

f)Even if the servo motor is equipped with an absolute position detection function, it is good practice to execute a zero return operation during initial system set up.

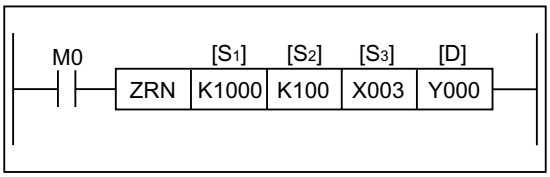


5.13.7 ZRN (FNC 156)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S_1	S_2	S_3	D	
ZRN FNC 156 Zero return	Return to zero home point after machine ON or initial setting.	K,H,KnX,KnY, KnM,KnS T,C,D,V,Z		X,Y, M,S	Y Note: ☒ Y000 or Y001 only	ZRN: 9 steps DZRN: 17 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Contents:

When executing incremental or absolute positioning, the PLC stores the current position values which increase or decrease during operation.

Using these values, the PLC always knows the machine position. However when the power to the PLC is turned off, this data is lost. To cope with this the machine should return to the zero point when the power is turned ON, or during initial set up, to teach the zero position.

[S_1] is the Zero Return Speed, [S_2] is the Creep Speed, [S_3] is the Near Point Signal, and [D] is the Pulse Output Designation.

Points to note:

a)Users may specify zero return speed [S_1] as, 16-bit 10 to 32,767Hz or 32-bit 10 to 100kHz.

b)Users may specify the creep speed [S_2} of 10 to 32,767Hz

c)If any device other than an input relay (X) is specified for the Near point signal [S3] it will be affected by the operation cycle of the PLC and the dispersion of the zero point may be large.

d)Only Y000 or Y001 can be used for the pulse output [D].
Because of the nature of the high speed output, transistor type output units should be used with this instruction. Relay type outputs will suffer a greatly reduced life, and will cause false outputs to occur.
To ensure a 'clean' output signal when using transistor type units, the load current should be 200mA or higher with the FX2N Series. The load current should be 10 - 100mA with the FX1S/ 1N Series. It may be necessary to use 'pull up' resistors.

e)If M8140 is set to ON, the clear signal is sent to the servo motor when the return to zero point is complete.

f)Related device numbers.

D8141 (upper digit) & D8140 (lower digit) : Current value register of Y000 (32-bit)
D8143 (upper digit) & D8142 (lower digit) : Current value register of Y001 (32-bit)
D8147 (upper digit) & D8146 (lower digit) : Maximum speed when FNC156, FNC158 or FNC159 are executed 100~100,000Hz.
D8148 : Acceleration/Deceleration time adopted when FNC156, FNC158 or FNC159 are executed.

M8145 : Y000 pulse output stop (immediate)
M8146 : Y001 pulse output stop (immediate)
M8147 : Y000 pulse output monitor (BUS/READY)
M8148 : Y001 pulse output monitor (BUS/READY)

g)When a Mitsubishi MR-H or MR-J2 servo amplifier equipped with absolute position detection function is used, the current position of the servo can be read by FNC 155 (ABS).

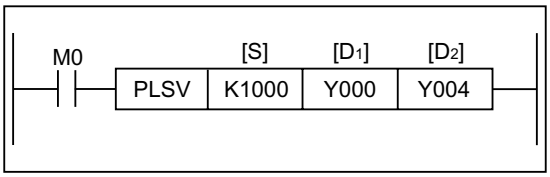
- Dog search function is not supported. Start zero return from the front side of the Near point signal.
- Attention should be paid to the instruction drive timing.

5.13.8 PLSV(FNC157)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D1	D2	
PLSV FNC 157 Pulse V	Variable speed pulse output	K,H, KnX,KnY, KnM,KnS T,C,D,V,Z	Y Note: ✉ Y000 or Y001 only	Y,M,S	PLSV 9 steps DPLSV 17 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

This is a variable speed output pulse instruction, with a rotation direction output.

[S] is the Pulse Frequency, [D_1] is the Pulse Output Designation, and [D_2] is the Rotation Direction Signal.

Points to note:

a)Users may use output pulse frequencies [S_1] of, 16-bit 10 to 32,767Hz or 32-bit 10 to 100kHz.

b)Only Y000 or Y001 can be used for the pulse output [D_1].

Because of the nature of the high speed output, transistor type output units should be used with this instruction. Relay type outputs will suffer a greatly reduced life, and will cause false outputs to occur.

To ensure a 'clean' output signal when using transistor type units, the load current should be 200mA or higher with the FX2N Series. The load current should be 10 - 100mA with the FX1S/ 1N Series. It may be necessary to use 'pull up' resistors.

c)Rotation direction signal output [D2} operated as follows: if [D_2] = OFF, rotation = negative, if [D_2] = ON, rotation = positive.

d)The pulse frequency [S] can be changed even when pulses are being output.

e)Acceleration/deceleration are not performed at start/stop. If cushion start/stop is required, increase or decrease the output pulse frequency [S] using the FNC67 RAMP instruction.

f)If the instruction drive contact turns off while pulses are output, the machine stops without deceleration

g)Once the instruction drive contact is off, re-drive of the instruction is not possible while the pulse output flag (Y000 : [M8147] Y001 : [M8148]) is ON.

h)The normal or reverse direction is specified by the positive or negative sign of the output pulse frequency [S]

i)Related device numbers.

D8141 (upper digit) & D8140 (lower digit) : Current value register of Y000 (32-bit)
D8143 (upper digit) & D8142 (lower digit) : Current value register of Y001 (32-bit)
M8145 : Y000 pulse output stop (immediate)
M8146 : Y001 pulse output stop (immediate)
M8147 : Y000 pulse output monitor (BUS/READY)
M8148 : Y001 pulse output monitor (BUS/READY)

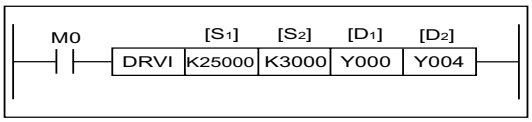
- Attention should be paid to the instruction drive timing.

5.13.9 DRVI (FNC 158)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S_1	S_2	D_1	D_2	
DRVI FNC 158 Drive to increment	Increment positioning	K,H, KnX,KnY, KnM,KnS T,C,D,V,Z		Y Note: ☒ Y000 or Y001 only	Y,M,S	DRVI 9 steps DDRVI 17 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

This instruction is for single speed positioning in the form of incremental movements.

$[S_1]$ is the Number of Pulses, $[S_2]$ is the Pulse Output Frequency, $[D_1]$ is the Pulse Output Designation, and $[D_2]$ is the Rotation Direction Signal.

Points to note:

a)The maximum number of pulses $[S_1]$ available are: 16-bit -32,768 to 32,767 pulses or 32-bit -999,999 to 999,999 pulses.

b)Users may use output pulse frequencies $[S_2]$, 16-bit 10 to 32,767Hz or 32-bit 10 to 100kHz.

c)Only Y000 or Y001 can be used for the pulse output $[D_1]$.
Because of the nature of the high speed output, transistor type output units should be used with this instruction. Relay type outputs will suffer a greatly reduced life, and will cause false outputs to occur.
To ensure a 'clean' output signal when using transistor type units, the load current should be 200mA or higher with the FX2N Series. The load current should be 10 - 100mA with the FX1S/ 1N Series. It may be necessary to use 'pull up' resistors.

d)Rotation direction signal output $[D_2]$ operated as follows: if $[D_2]$ = OFF, rotation = negative, if $[D_2]$ = ON, rotation = positive.

e)If the contents of an operand are changed while the instruction is executed, it is not reflected on the operation. The new contents become effective when the instruction is next driven.

f)If the instruction drive contact turns off while the instruction is being executed, the machine decelerates and stops. At this time the execution complete flag M8029 does not turn ON.

g)Once the instruction drive contact is off, re-drive of the instruction is not possible while the pulse output flag (Y000 : [M8147], Y001 : [M8148]) is ON.

h)For operation in the incremental drive method, the travel distance from the current position is specified with either a positive or a negative symbol.

i)The minimum value of output pulse frequency which can be actually used is determined by the following equation

$$\sqrt{\text{MaxSpeed[D8147,D8146]Hz} \div (2 \times (\text{Acceleration\backslash Deceleration[D8148]ms} \div 1000))}$$

f)Related device numbers.

D8145 : Bias speed adopted when either FNC158, DRVI or FNC159, DRVA are executed
D8147 (upper digit) & D8146 (lower digit) : Maximum speed when FNC156, FNC158 or FNC159 are executed 100~100,000Hz.
D8148 : Acceleration/Deceleration time adopted when FNC156, FNC158 or FNC159 are executed.
M8145 : Y000 pulse output stop (immediate)
M8146 : Y001 pulse output stop (immediate)
M8147 : Y000 pulse output monitor (BUS/READY)
M8148 : Y001 pulse output monitor (BUS/READY)

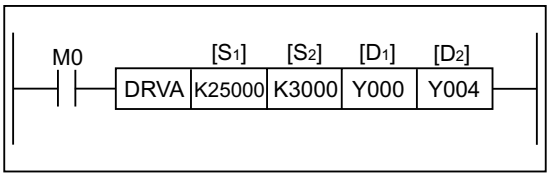
- Attention should be paid to the instruction drive timing.

5.13.10 DRVA(FNC 159)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S_1	S_2	D_1	D_2	
DRVA FNC 159 Drive to absolute	Absolute positioning	K,H, KnX,KnY, KnM,KnS T,C,D,V,Z		Y Note: ☒ Y000 or Y001 only	Y,M,S	DRVA 9 steps DDRVA 17 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

This instruction is for single speed positioning using a zero home point and absolute measurements.

$[S_1]$ is the Number of Pulses, $[S_2]$ is the Output Frequency, $[D_1]$ is the Pulse Output Designations, and $[D_2]$ is the Rotation Direction Signal.

Points to note:

a)The target position for absolute positioning $[S_1]$ can be: 16-bit -32,768 to 32,767 pulses or 32-bit -999,999 to 999,999 pulses.

b)Users may use output pulse frequencies $[S_2]$, 16-bit 10 to 32,767Hz or 32-bit 10 to 100kHz.

c)Only Y000 or Y001 can be used for the pulse output $[D_1]$.
Because of the nature of the high speed output, transistor type output units should be used with this instruction. Relay type outputs will suffer a greatly reduced life, and will cause false outputs to occur.
To ensure a 'clean' output signal when using transistor type units, the load current should be 200mA or higher with the FX2N Series. The load current should be 10 - 100mA with the FX1S/ 1N Series. It may be necessary to use 'pull up' resistors.

d)Rotation direction signal output $[D_2]$ operated as follows: if $[D_2]$ = OFF, rotation = negative, if $[D_2]$ = ON, rotation = positive.

e)If the contents of an operand are changed while the instruction is executed, it is not reflected on the operation. The new contents become effective when the instruction is next driven.

f)If the instruction drive contact turns off while the instruction is being executed, the machine decelerates and stops. At this time the execution complete flag M8029 does not turn ON.

g)Once the instruction drive contact is off, re-drive of the instruction is not possible while the pulse output flag (Y000 : [M8147], Y001 : [M8148]) is ON.

h)For operation in the absolute drive method, the travel distance from the zero point is specified.

i)The minimum value of output pulse frequency which can be actually used is determined by the following equation

$$\sqrt{\text{MaxSpeed[D8147,D8146]Hz} \div (2 \times (\text{Acceleration\backslash Deceleration[D8148]ms} \div 1000))}$$

f)Related device numbers.

D8145 : Bias speed adopted when either FNC158, DRVI or FNC159, DRVA are executed
D8147 (upper digit) & D8146 (lower digit) : Maximum speed when FNC156, FNC158 or FNC159 are executed 100~100,000Hz.
D8148 : Acceleration/Deceleration time adopted when FNC156, FNC158 or FNC159 are executed.
M8145 : Y000 pulse output stop (immediate)
M8146 : Y001 pulse output stop (immediate)
M8147 : Y000 pulse output monitor (BUS/READY)
M8148 : Y001 pulse output monitor (BUS/READY)

- Attention should be paid to the instruction drive timing.

Memo

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.14 Real Time Clock Control - FNC 160 to FNC 169

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

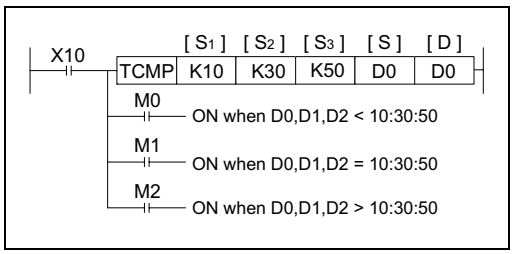
✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.14.1 TCMP (FNC 160)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands					Program steps
		S_1	S_2	S_3	S	D	
TCMP FNC 160 (Time Compare)	Compares two times - results of <, = and > are given	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z			T, C, D Note: 3 consecutive devices are used.	Y, M, S	TCMP, TCMPP: 11 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Contents:

S_1, S_2 and S_3 represent hours, minutes and seconds respectively. This time is compared to the time value in the 3 data devices specified by the head address S. The result is indicated in the 3 bit devices specified by the head address D.

The bit devices in D indicate the following:

D_{+0} is set ON, when the time in S is less than the time in S_1, S_2 and S_3.
D_{+1} is set ON, when the time in S is equal to the time in S_1, S_2 and S_3.
D_{+2} is set ON, when the time in S is greater than the time in S_1, S_2 and S_3.

Points to note:

a) The status of the destination devices is kept, even if the TCMP instruction is deactivated.

b) The comparison is based on the time value specified in the source devices.
- The valid range of values for S1 and S+0 is 0 to 23 (Hours).
- The valid range of values for S2 and S+1 is 0 to 59 (Minutes).
- The valid range of values for S3 and S+2 is 0 to 59 (Seconds).

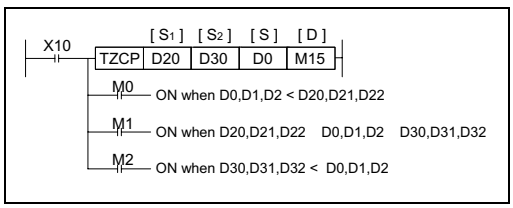
c) The current time of the real time clock can be compared by specifying D8015 (Hours), D8014 (Minutes) and D8013 (Seconds) as the devices for S_1, S_2 and S_3 respectively.

5.14.2 TZCP (FNC 161)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands				Program steps
		S1	S2	S	D	
TZCP FNC 161 (Time Zone Compare)	Compares a time to a specified time range - results of <, = and > are given	T, C, D S1 must be less than or equal to S2.			Y, M, S	TZCP, TZCPP: 9 steps
		Note: 3 consecutive devices are used for all				

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Contents:

S1, S2 and S represent time values. Each specifying the head address of 3 data devices. S is compared to the time period defined by S1 and S2. The result is indicated in the 3 bit devices specified by the head address D.

The bit devices in D indicate the following:

D+0 is set ON, when the time in S is less than the times in S1 and S2.
D+1 is set ON, when the time in S is between the times in S1 and S2.
D+2 is set ON, when the time in S is greater than the times in S1 and S2.

Points to note:

a) The status of the destination devices is kept, even if the TCMP instruction is deactivated.

b) The comparison is based on the time value specified in the source devices.

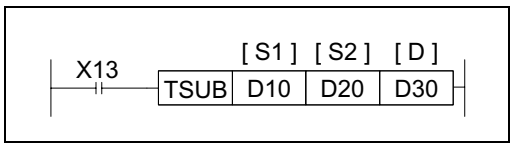
- The valid range of values for S1 and S+0 is 0 to 23 (Hours).
- The valid range of values for S2 and S+1 is 0 to 59 (Minutes).
- The valid range of values for S3 and S+2 is 0 to 59 (Seconds).

5.14.3 TADD (FNC 162)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
TADD FNC 162 (Time Addition)	Adds two time values together to give a new time	T, C, D Note: 3 consecutive devices are used to represent hours, minutes and seconds respectively.			TADD, TADDP: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Zero M8020 Carry M8022



Contents:

Each of S1, S2 and D specify the head address of 3 data devices to be used a time value.
The time value in S1 is added to the time value in S2, the result is stored to D as a new time value.

Points to note:

a) The addition is performed according to standard time values. Hours, minutes and seconds are kept within correct limits. Any overflow is correctly processed.

S1		S2		D
D10: 10 hours		D20: 30 hours		D30: 13 hours
D11: 30 mins	+	D21: 10 mins	=	D31: 41 mins
D12: 27 secs		D22: 49 secs		D32: 16 secs
10:30:29		03:10:49		13:41:16

b) If the addition of the two times results in a value greater than 24 hours, the value of the result is the time remaining above 24 hours.

S1		S2		D
D10: 10 hours		D20: 18 hours		D30: 13 hours
D11: 17 mins	+	D21: 12 mins	=	D31: 41 mins
D12: 29 secs		D22: 34 secs		D32: 16 secs
10:17:29		18:12:34		04:30:03
				M8022 ON

When this happens the carry flag M8022 is set ON.

c) If the addition of the two times results in a value of zero (0:00:00: 0 hours, 0 minutes, 0 seconds) then the zero flag M8020 is set ON.

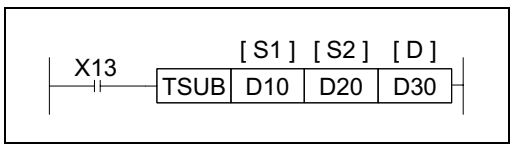
d) The same device may be used as a source (S1 or S2) and destination device. In this case the addition is continually executed; the destination value changing each program scan. To prevent this from happening, use the pulse modifier or an interlock program.

5.14.4 TSUB (FNC 163)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
TSUB FNC 163 (Time Subtrac-tion)	Subtracts one time value from another to give a new time	T, C, D Note: 3 consecutive devices are used.			TSUB, TSUBP: 7 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P	Flags	Zero M8020 Borrow M8021



Contents:

Each of S_1, S_2 and D specify the head address of 3 data devices to be used a time value.
The time value in S_1 is subtracted from the time value in S_2, the result is stored to D as a new time value.

Points to note:

a) The subtraction is performed according to standard time values. Hours, minutes and seconds are kept within correct limits. Any underflow is correctly processed.

S_1		S_2		D
D10: 10 hours		D20: 3 hours		D30: 7 hours
D11: 30 mins	-	D21: 10 mins	=	D31: 19 mins
D12: 27 secs		D22: 49 secs		D32: 38 secs
10:30:27		03:10:49		07:19:38

b) If the subtraction of the two times results in a value less than 00:00:00 hours, the value of the result is the time remaining below 00:00:00 hours.

S_1		S_2		D
D10: 10 hours		D20: 18 hours		D30: 13 hours
D11: 17 mins	-	D21: 12 mins	=	D31: 41 mins
D12: 29 secs		D22: 34 secs		D32: 16 secs
10:17:29		18:12:34		16:04:55
				M8021 ON

When this happens the borrow flag M8021 is set ON.

c) If the subtraction of the two times results in a value of zero (00:00:00 hours) then the zero flag M8020 is set ON.

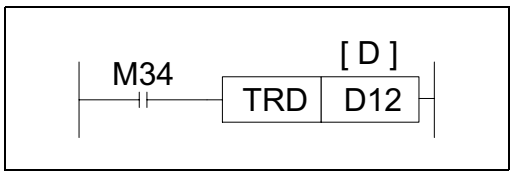
d) The same device may be used as a source (S1 or S2) and destination device. In this case the subtraction is continually executed; the destination value changing each program scan. To prevent this from happening, use the pulse modifier or an interlock program.

5.14.5 TRD (FNC 166)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		D	
TRD FNC 166 (Time Read)	Reads the current value of the real time clock to a group of registers	T, C, D Note: 7 consecutive devices are used.	TRD, TRDP: 5 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Contents:

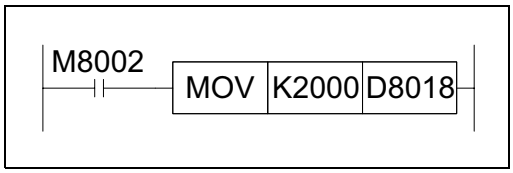
The current time and date of the real time clock are read and stored in the 7 data devices specified by the head address D.

The 7 devices are set as follows:

Device	Meaning	Values		Device	Meaning
D8018	Year	00-99	⇒	D+0	Year
D8017	Month	01-12	⇒	D+1	Month
D8016	Date	01-31	⇒	D+2	Date
D8015	Hours	00-23	⇒	D+3	Hours
D8014	Minutes	00-59	⇒	D+4	Minutes
D8013	Seconds	00-59	⇒	D+5	Seconds
D8019	Day	0-6 (Sun-Sat)	⇒	D+6	Day

Points to note:

The year is read as a two digit number. This can be change to a 4 digit number by setting D8018 to 2000 during the first program scan; see following program extract.



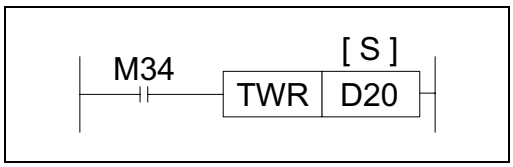
If this is done then the clock year should not be used during the first scan as it will be a two digit number before the instruction and a value of 2000 after the instruction until the END instruction executes. After the first scan the year is read and written as a 4 digit number.

5.14.6 TWR (FNC 167)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands	Program steps
		S	
TWR FNC 167 (Time Write)	Sets the real time clock to the value stored in a group of registers	T, C, D Note: 7 consecutive devices are used.	TWR, TWRP: 5 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Contents:

The 7 data devices specified with the head address S are used to set a new current value of the real time clock.

The seven devices

Device	Meaning	Values		Device	Meaning
S+0	Year	00-99	⇒	D8018	Year
S+1	Month	01-12	⇒	D8017	Month
S+2	Date	01-31	⇒	D8016	Date
S+3	Hours	00-23	⇒	D8015	Hours
S+4	Minutes	00-59	⇒	D8014	Minutes
S+5	Seconds	00-59	⇒	D8013	Seconds
S+6	Day	0-6 (Sun-Sat)	⇒	D8019	Day

Points to note:

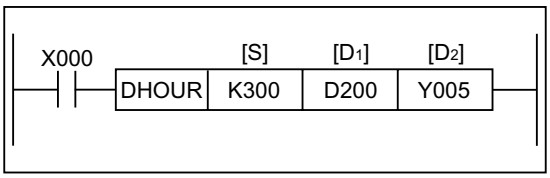
This instruction removes the need to use M8015 during real time clock setting. When setting the time it is a good idea to set the source data to a time a number of minutes ahead and then drive the instruction when the real time reaches this value.

5.14.7 Hour (FNC 169)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands			Program steps
		S	D_1	D_2	
Hour FNC 169 Hour meter	Hour meter	K,H, KnX, KnY, KnM, KnS, T,C,D,V,Z	D Note: Data register should be battery backed	Z,Y, M,S	

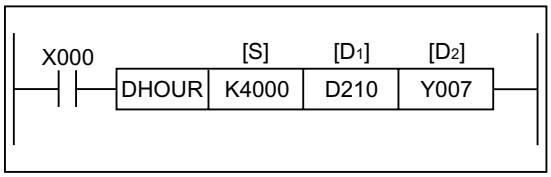
16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation 1: 16 bit instruction

[S] = Period of time before [D_2] turns on (Hrs)
[D_1] = Current value in Hours
[D_1]+1 = Current value, if less than 1 hour, time is specified in seconds.
[D_2] = Alarm output destination, turns on when [D_1] exceeds [S]

In the above example, [D_2] turns on at 300 hours and 1 second.



Operation 2: 32 bit instruction

[S] = Period of time in which [D_2] turns on (Hrs)
[D_1] = Current value in Hours
[D_1]+2 = Current value, if less than 1 hour. In seconds
[D_2] = Alarm output destination, when [D_1] exceeds [S]

In the above example, [D_2] turns on at 4000 hours and 1 second.

Points to note:

a)In order to continuously use the current value data, even after a power OFF and ON, specify a data register which is backed up against power interruption.

b)The hour meter will continue operation even after the alarm output [D_2] turns ON.
Operation will stop when the value of [D_1] reaches the maximum for the specified 16 or 32 bit operation.
If continuous operation id required, clear the value stored in [D_1] to [D_1]+1 (16-bit) and [D_1] to [D_1]+2 (32-bit).

Memo

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.15 Gray Codes - FNC 170 to FNC 179

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

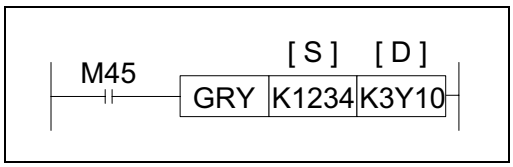
✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.15.1 GRY (FNC 170)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
GRY FNC 170 (Gray Code)	Calculates the gray code value of an integer	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	GRY,GRYP: 5 steps DGRY,DGRYP 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The binary integer value in S is converted to the GRAY CODE equivalent and stored at D.

Points to Note:

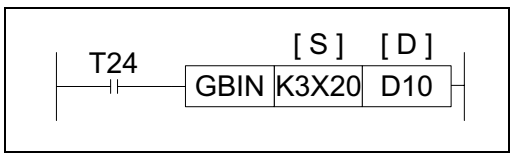
The nature of gray code numbers allows numeric values to be quickly output without the need for a strobing signal. For example, if the source data is continually incremented, the new output data can be set each program scan.

5.15.2 GBIN (FNC 171)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
GBIN FNC 171 (Gray Code)	Calculates the integer value of a gray code	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	GBIN,GBINP: 5 steps DGBIN, DGBINP: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The GRAY CODE value in S is converted to the normal binary equivalent and stored at D.

Points to Note:

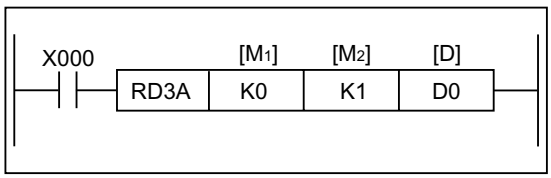
This instruction can be used to read the value from a gray code encoder.
If the source is set to inputs X0 to X17 it is possible to speed up the reading time by adjusting the refresh filter with FNC 51 REFF.

5.15.3 RD3A (FNC 176)

FX1S | FX1N | FX2N | FX2NC

Mnemonic	Function	Operands			Program steps
		M1	M2	D	
RD3A FNC 176 Read analog block	Analog block read for FX0N-3A, FX2N-2AD	K,H. KnX, KnY,KnM,KnS, T,C,D,V,Z		KnY, KnM,KnS T.C,D,V,Z	RD3A 7 steps

16 BIT OPERATION | 32 BIT OPERATION | PULSE-P



Operation:

This instruction reads the analog input value of the FX0N-3A or FX2N-2AD block.

[M_1] = Special block number, K0 to K7
[M_2] = Analog input channel number, K1/K21 or K2/K22
[D] = Read data

Points to note:

Adjustment of the FX0N-3A or FX2N-2AD analog block characteristics should be completed in advance of using this instruction. For guidance please see the relevant manual.

For the FX0N-3A, K1 = Channel 1, K2 = Channel 2. D = 0~255(8 bit)
For the FX2N-2AD, K21 = Channel 1, K22 = Channel 2 D = 0~4095(12bit)

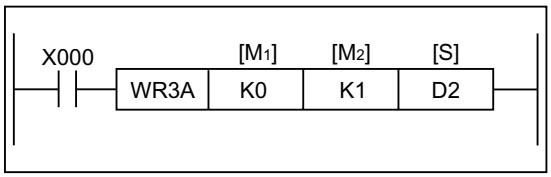
FX1N series can only communicate with the FX0N-3A

5.15.4 WR3A (FNC 177)

FX1S | FX1N | FX2N | FX2NC

Mnemonic	Function	Operands			Program steps
		M1	M2	S	
WR3A FNC 177 Write to Analog block	Write data to analog block FX0N-3A, FX2N-2AD, FX2N-2DA	K,H. KnX, KnY,KnM,KnS, T,C,D,V,Z		KnY, KnM,KnS T.C,D,V,Z	WR3A 7 steps

16 BIT OPERATION | 32 BIT OPERATION | PULSE-P



Operation:

This instruction writes data to the FX0N-3A or FX2N-2DA analog block.

[M_1] = Special block number, K0 to K7
[M_2] = Analog output channel number, K1/K21 or K22
[S] = Write data

Points to note:

Adjustment of the FX0N-3A or FX2N-2DA analog block characteristics should be completed in advance of using this instruction. For guidance please see the relevant manual.

For the FX0N-3A, K1 = Channel 1 (only 1 output channel available) S = 0~255(8 bit)
For the FX2N-2DA, K21 = Channel 1, K22 = Channel 2 S = 0~4095(12bit)

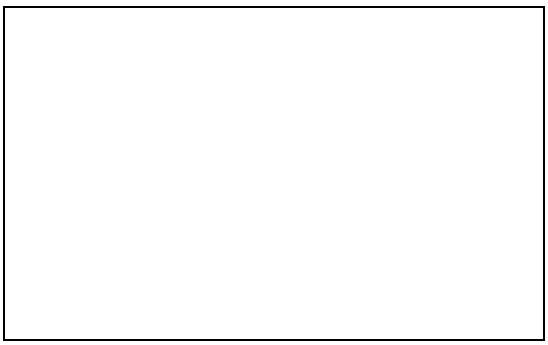
FX1N series can only communicate with the FX0N-3A

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.16 Additional Functions - FNC 180 to FNC 189

Contents:



Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆ - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

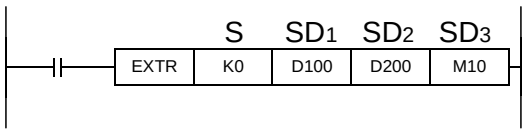
✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.16.1 EXTR (FNC 180)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	SD1 SD2 SD3	
EXTR FNC 180 (External ROM)	External ROM instruction, execution commands	K, H,	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z, X, Y, M, S.	EXTR 9 steps DEXTR, DEXTRP 17 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The value of S stored in the extension ROM (K0 to K32767) defines the function number and the instruction.
SD1, SD2 and SD3 are parameters of the application instruction. S or D varies depending on the function number. The type of operation (16 bit, 32 bit, pulse) is determined from the instruction number.

Points to Note:

In some function numbers, the parameters SD1 to SD3 may not be required due to specifications. In such a case K0 should be written in the program. K0 is ignored in the internal processing of the PLC.

As each of the external ROM cassettes (FX2N-ROM-E1 and FX2NC-ROM-CE1) attach to the memory port of an FX2N or FX2NC series PLC, both are equipped with a 16K step EEPROM. In addition, the FX2NC-ROM-CE1 also contains a real-time clock.
Accordingly, the FX2N-ROM-E1 and FX2NC-ROM-CE1 are compatible as advanced units of the FX-EEPROM-16 and FX2NC-EEPROM16C respectively.

The FX2N-ROM-E1 and FX2NC-ROM-CE1 are only operable with FX2N and FX2NC units of V3.00 or later

5.16.1.1 Inverter Communication

External ROM cassette functions 10 to 13 are for reading and writing data to/from an inverter using signal instructions. These functions are available when an FX2N-485-BD or FX0N-485ADP is attached to the PLC, for communication with a Mitsubishi Electric A500/E500/S500 series inverter.

Function No.	Function	Data Direction	Reference to Inverter manual
EXTR K10	Operation monitoring	INV to PLC	Execute operation control as for computer link, and refer to 'monitoring' for communication functions.
EXTR K11	Operation Control	PLC to INV	
EXTR K12	Parameter read	INV to PLC	Refer to the parameter code list in the relevant manuals appendix.
EXTR K13	Parameter write	PLC to INV	

5.16.1.1.1 Restrictions

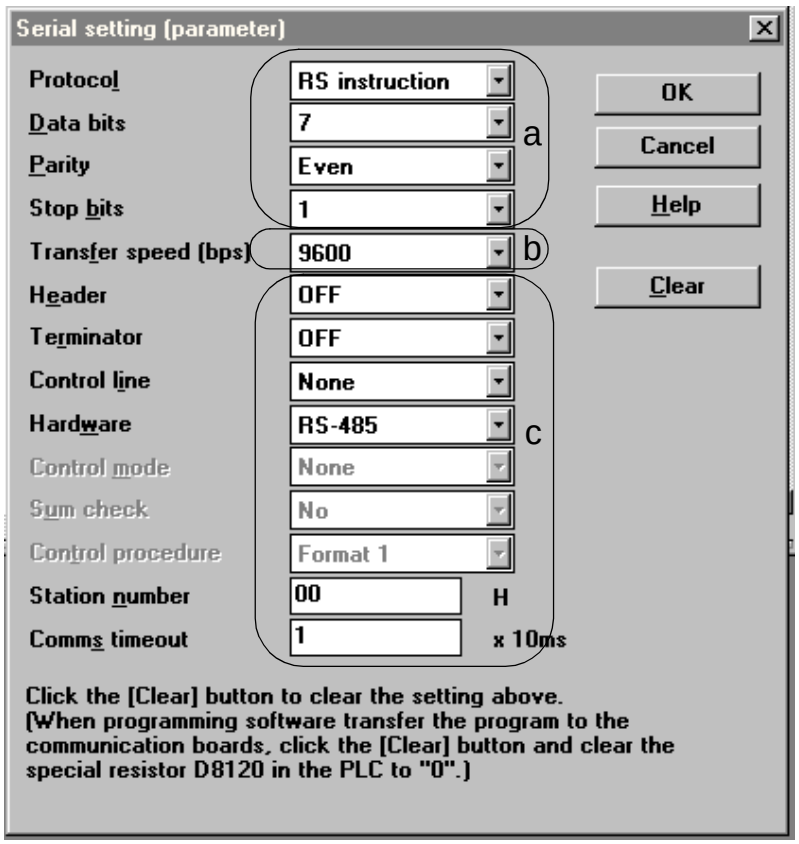
Six digit commands that are supported in the E500 and S500 series inverters are not supported by the EXTR function.

5.16.1.1.2 Settings in the PLC

EXTR K10 to K13 use the FX2N-485-BD or FX0N-485ADP in the same way as the RS instruction (FNC 80). The communication conditions should be set in the 'Serial setting' parameter in FX-PCS/WIN-E or GX Developer.

When setting the Serial Parameters from the software packages FX-PCS/WIN-E or GX Developer, do not use data registers D8120, D8121, and D8129 in the user program. These registers are set by the software package and if changed in the user program will cause communication problems between the PLC and the Inverter.

For FX-PCS/WIN-E:
1) Select "Option" - "Serial setting (Parameter)"
2) Click "Yes"
3) Set "Serial setting (Parameters)" as shown below



a) Set these parameters as show on the left.
DO NOT select "Link"

b) Select either 19200, 9600 or 4800. This value should be the same as set in the parameter of the inverter.

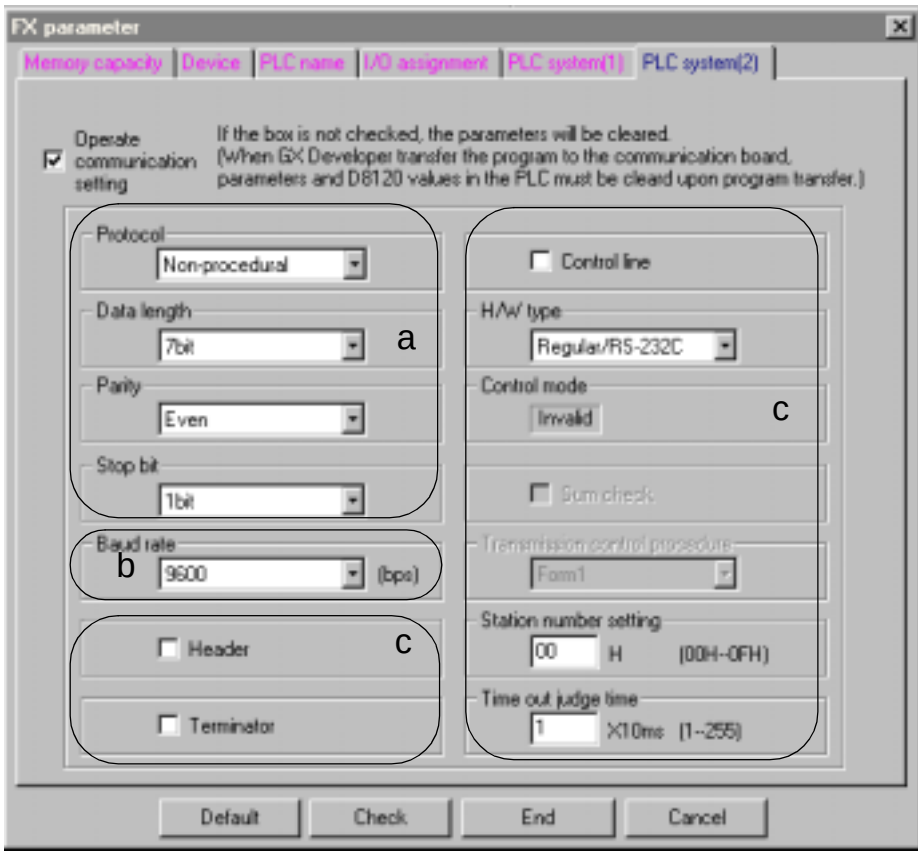
c) These parameters do not affect communication with the inverter.

For GX Developer

1) Select “Parameters”

2) Select “PLC Parameter”

3) Select “PLC System (2)” and set as shown below.



a) Set these parameters as show on the left.
DO NOT select “Link”

b) Select either 19200, 9600 or 4800. This value should be the same as set in the inverter.

c) These parameters do not affect communication with the inverter.

5.16.1.1.3 Inverter settings and PLC communication settings

Inverter communication specification and application to PLC

			Inverter specifications	Application to PLC
Transmission standard			RS-485	RS-485
Number of connected units			1:N (8 units maximum)	1:N (8 units maximum)
Communication speed			19200, 9600 or 4800 bps (selectable)	19200, 9600 or 4800 bps (selectable)
Control procedure			Asynchronous system	Asynchronous system
Communication method			Half duplex	Half duplex
Communication specifications	Character type		ASCII (7 or 8 bits) (selectable)	Fixed to 7 bits
	Stop bit length		1 or 2 bits (selectable)	Fixed to 1 bit
	Terminator		CR/LF (absence/presence selectable)	Fixed to CR only
	Check method	Parity check/	Fixed (even), fixed (odd) or not fixed (selectable)	Fixed to even parity
		Sum check	Fixed	Fixed
	Waiting time setting		Set by customer	Set by communication data

Note:
Some of the specifications above are fixed in the PLC but variable in the inverter. This has been done to ease set up and reduce any possible problems during configuration.

For more information on the inverter, please see the appropriate inverter manual.

A500 series settings

Parameter number	Description	Set value	Contents	Setting for comms to PLC
117	Station No.	0~31	Corresponds to the station No. specified from the PU connector. If two or more inverters, set the station No. at each inverter	Align setting with station number in the sequence program
118	Communication speed	48	4800bps	Normally select 192. If high speed processing in PLC use 96 or 48
		96	9600bps	
		192	19200bps	
119	Stop bit length / Data bit length	0	8 data bits / 1 stop bit	Select 10
		1	8 data bits / 2 stop bits	
		10	7 data bits / 1 stop bit	
		11	7 data bits / 2 stop bits	
120	Parity check	0	Absent	Select 2
		1	Present (Odd)	
		2	Present (Even)	
121	Number of communication retries	0,1~10	Set number of retries after data receive error. If this value is exceeded, inverter comes to alarm stop	During trial run select 9999 and perform adjustment. During actual operation select value in accordance with system specifications
		9999 (65535)	If a comms error occours, inverter will not come to alarm stop. At this time inverter can be coasted to stop by MRS or RES input. During comms. error, the LF signal is output to the open collector output. Allocate one terminal from Pr.190-195	
122*	Communication check time interval	0	Comms not executed	While default value 0 is selected, communications are disabled. Select 9999, perform adjustment, then select optimal value
		0.1~999.8	Set comms check time interval. If a no-communication state persists for longer than the set time, inverter comes to alarm stop. See Note on page 195.	
		9999	Comms check suspended	
123	Waiting time setting	0~150	Set waiting time between data transmission and response	Select 9999
		9999	Waiting time set by communication data	
124	CR,LF selection	0	CR & LF instructions absent	Select 1
		1	CR instruction present	
		2	CR & LF instructions present	
342	E^2PROM write selection	0	Parameter write from computer to EEPROM	Select either value according to system specifications
		1	Parameter write from computer to RAM	

***A500 + A5NR settings**

Parameter number	Name	Setting range	Setting increment	Inverter station No.	Setting for comms to PLC
331	Inverter station No.	0~31	1	0	Align setting with station number in the sequence program
332	Comms speed	3,6,12,24,48,96,192	1	96	Normally select 192. If high speed processing in PLC use 96 or 48
333	Stop bit length	0,1 (8 bit) 10,11 (7bit)	1	1	Select 10
334	Parity check yes/no	0,1,2	1	2	Select 2
335	Comms retry count	0~10, 9999	1	1	During trial run select 9999 and perform adjustment. During actual operation select value in accordance with system specifications
336	Comms check time interval	0~999.8, 9999	0.1	0	
337*	Wait time setting	0~150ms, 9999	1ms	9999	While default value 0 is selected, communications are disabled. Select 9999, perform adjustment, then select optimal value. See Note below.
338	Operation command write	0,1	1	0	Select either value in accordance to system specifications
339	Speed command write	0,1	1	0	Select either value in accordance to system specifications
340	Link start mode selection	0,1,2	1	0	Select either value in accordance to system specifications
341	CR/LF yes/no selection	0,1,2	1	1	Select 1 CR only
342	EEPROM write selection	0: write to EEPROM 1: write to RAM	1	0	Select either value in accordance to system specifications

*Note:
The time settings should be set as low as possible to avoid problems during a communication failure. The inverter will continue to run during the set time which may cause equipment damage or raise a safety issue. Please turn the inverter Off when communication problems are encountered.

E500 series settings

Parameter number	Description	Set value	Contents	Setting for comms to PLC
117	Station No.	0~31	Corresponds to the station No. specified from the PU connector. If two or more inverters, set the station No. at each inverter	Align setting with station number in the sequence program
118	Communication speed	48	4800bps	Normally select 192. If high speed processing in PLC use 96 or 48
		96	9600bps	
		192	19200bps	
119	Stop bit length / Data bit length	0	8 data bits / 1 stop bit	Select 10
		1	8 data bits / 2 stop bits	
		10	7 data bits / 1 stop bit	
		11	7 data bits / 2 stop bits	
120	Parity check	0	Absent	Select 2
		1	Present (Odd)	
		2	Present (Even)	
121	Number of communication retries	0,1~10	Set number of retries after data receive error. If this value is exceeded, inverter comes to alarm stop	During trial run select 9999 and perform adjustment. During actual operation select value in accordance with system specifications
		9999 (65535)	If a comms error occours, inverter will not come to alarm stop. At this time inverter can be coasted to stop by MRS or RES input. During comms. error, the LF signal is output to the open collector output. Allocate one terminal from Pr.190-192	
122*	Communication check time interval	0	Comms not executed	While default value 0 is selected, communications are disabled. Select 9999, perform adjustment, then select optimal value
		0.1~999.8	Set comms check time interval. If a no-communication state persists for longer than the set time, inverter comes to alarm stop. See Note on page 195.	
		9999	Comms check suspended	
123	Waiting time setting	0~150	Set waiting time between data transmission and response	Select 9999
		9999	Waiting time set by communication data	
124	CR,LF selection	0	CR & LF instructions absent	Select 1
		1	CR instruction present	
		2	CR & LF instructions present	

S500 series settings

Parameter number	Description	Set value	Contents	Setting for comms to PLC
n1	Station No.	0~31	Corresponds to the station No. specified from the PU connector. If two or more inverters, set the station No. at each inverter	Align setting with station number in the sequence program
n2	Communication speed	48	4800bps	Normally select 192. If high speed processing in PLC use 96 or 48
		96	9600bps	
		192	19200bps	
n3	Stop bit length / Data bit length	0	8 data bits / 1 stop bit	Select 10
		1	8 data bits / 2 stop bits	
		10	7 data bits / 1 stop bit	
		11	7 data bits / 2 stop bits	
n4	Parity check	0	Absent	Select 2
		1	Present (Odd)	
		2	Present (Even)	
n5	Number of communication retries	0,1~10	Set number of retries after data receive error. If this value is exceeded, inverter comes to alarm stop	During trial run select 9999 and perform adjustment. During actual operation select value in accordance with system specifications
		9999 (65535)	If a comms error occours, inverter will not come to alarm stop. At this time inverter can be coasted to stop by MRS or RES input. During comms. error, the LF signal is output to the open collector output. Allocate one terminal from Pr.64-65	
n6*	Communication check time interval	0	Comms not executed	While default value 0 is selected, communications are disabled. Select 9999, perform adjustment, then select optimal value
		0.1~999.8	Set comms check time interval. If a no-communication state persists for longer than the set time, inverter comes to alarm stop. See Note on Page 195.	
		9999	Comms check suspended	
n7	Waiting time setting	0~150	Set waiting time between data transmission and response	Select 9999
		9999	Waiting time set by communication data	
n11	CR,LF selection	0	CR & LF instructions absent	Select 1
		1	CR instruction present	
		2	CR & LF instructions present	

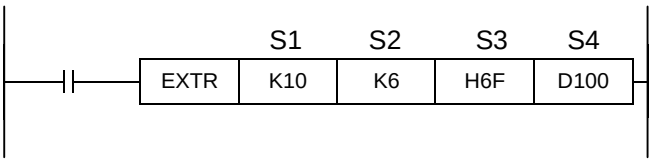
Example of transmission format when data is written from PLC to inverter

ENQ	Inverter station 6		Command H80		Wait time =0	Data = 1234				SUM		CR
H05	H30	H36	H38	H30	H30	H31	H32	H33	H34	H43	H38	H0D

H30+H36+H38+H30+H30+H31+H32+H33+H34=H1C8

C=H43 8=H38

5.16.1.1.4 EXTR K10 - Monitoring operations (Inverter to PLC)



16 Bit Operation	✓
32 Bit Operation	×
Pulse - P	×

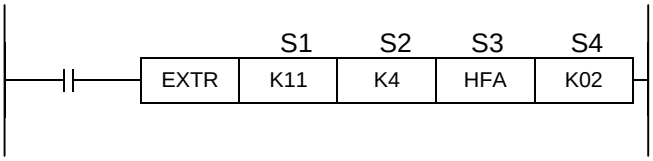
Parameter	Device type	Parameter range
S1	K/H	K10: function No. to monitor inverter operations
S2	KH, D	Inverter station number (0 to 31)
S3	KH, D	Inverter instructuion code (varies with model)
S4	D, KnY, KnM, KnS	Read value storage destination

Details of S3

Instruction Code	Monitor contents	A500	E500	S500	Data digits	Comms format
H7B	Operation mode	✓	✓	✓	4	B⇨E |F
H6F	Output frequency	✓	✓	✓	4	B⇨E |F
H70	Output current	✓	✓	✓	4	B⇨E |F
H71	Output voltage	✓	✓		4	B⇨E |F
H72	Special monitor	✓			4	B⇨E |F
H73	Special monitor selection No.	✓			2	B⇨E' |F
H74	Alarm definition	✓	✓	✓	4	B⇨E |F
H75	Alarm definition	✓	✓	✓	4	B⇨E |F
H76	Alarm definition	✓	✓		4	B⇨E |F
H77	Alarm definition	✓	✓		4	B⇨E |F
H7A	Inverter status monitor	✓	✓	✓	2	B⇨E' |F
H6E	Set frequency read (EEPROM)			✓	4	B⇨E |F
H6D	Set frequency read (RAM)			✓	4	B⇨E |F
H00~H7B	Parameter read	✓	✓	✓	4	B⇨E |F
H7F	Link parameter	✓	✓	✓	2	B⇨E' |F
H6C	Second parameter change	✓	✓	✓	2	B⇨E' |F

Note:
The shaded area is supported but is executed by EXTR K12.

5.16.1.1.5 EXTR K11 - Control operations (PLC to Inverter)



16 Bit Operation	✓
32 Bit Operation	×
Pulse - P	×

Parameter	Device type	Parameter range
S1	K/H	K11: function No. to control inverter operations
S2	KH, D	Inverter station number (0 to 31)
S3	KH, D	Inverter instructuion code (varies with model)
S4	KH D, KnX, KnY, KnM, KnS	Value to be written to inverter

Details of S3

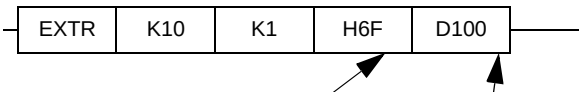
S3	Control contents	A500	E500	S500	Data digits	Comms format
HFB	Operation mode	✓	✓	✓	4	A⇨C |D
HF3	Special monitor selection No.	✓			2	A'⇨C |D
HFA	Operation command	✓	✓	✓	2	A'⇨C |D
HEE	Set frequency read (EEPROM)	✓	✓	✓	4	A⇨C |D
HED	Set frequency read (RAM)	✓	✓	✓	4	A⇨C |D
HFD	Inverter reset	✓	✓	✓	4	A (no response)
HF4	Alarm definition batch clear	✓		✓	4	A⇨C |D
HFC	All parameter clear	✓	✓	✓	4	A⇨C |D
H80~HFD	Parameter write	✓	✓	✓	4	A⇨C |D
HFC	User clear	✓			4	A⇨C |D
HFF	Link parameter extension setting	✓	✓	✓	2	A'⇨C |D
HEC	Second parameter changing (code FF=1)	✓	✓	✓	2	A'⇨C |D

Note:
The shaded areas are re-written in the internal processing when EXTR K13 is executed.

5.16.1.1.6 Relationship between EXTR K10/K11 and A500/E500/S500 series

The page below is taken from section 4.2.41, paragraph 5 “Instructions for the program” <setting items and set data> in the FR A500 series instruction manual, IB(NA)-66790-G

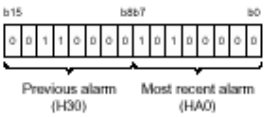
The following example reads the Output Frequency from Inverter station #1 and stores this value to D100.



No.	Item		Instruction Code	Description	Number of Data Digits
1	Operation mode	Read	H7B	H0000: Communication option operation H0001: External operation H0002: Communication operation (PU connector)	4 digits
		Write	HFB	H0000: Communication option operation H0001: External operation H0002: Communication operation (PU connector)	
2 Monitoring	Output frequency [speed]		H6F	H0000 to HFFFF: Output frequency (hexadecimal) in 0.01Hz increments (Speed (hexadecimal) in r/min increments if Pr. 37 = 1 to 9998 or Pr. 144 = 2 to 10, 102 to 110.)	4 digits
	Output current		H70	H0000 to HFFFF: Output current (hexadecimal) in 0.01A increments	4 digits
	Output voltage		H71	H0000 to HFFFF: Output voltage (hexadecimal) in 0.1V increments	4 digits
	Special monitor		H72	H0000 to HFFFF: Monitored data selected by instruction code HF3	4 digits
	Special monitor selection No.	Read	H73	H01 to H0E Monitor selection data	2 digits
		Write	HF3		
	Alarm definition		H74 to H77	H0000 to HFFFF: Two most recent alarm definitions Read data: [Example] H30A0 (Previous alarm ..... THT) (Most recent alarm ..... OPT)	4 digits

Monitor selection data

Data	Description	Increments	Data	Description	Increments
H01	Output frequency	0.01Hz	H09	Regenerative brake	0.1%
H02	Output current	0.01A	H0A	Electronic overcurrent protection load factor	0.1%
H03	Output voltage	0.1V	H0B	Output current peak value	0.01A
H05	Frequency setting	0.01Hz	H0C	Converter output voltage peak value	0.1V
H06	Running speed	1 r/min	H0D	Input power	0.01kW
H07	Motor torque	0.1%	H0E	Output power	0.01kW

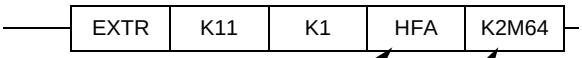


Alarm data

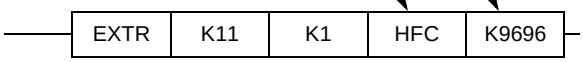
Data	Description	Data	Description	Data	Description
H00	No alarm	H80	GF	HD5	MB1
H10	0C1	H81	LF	HD6	MB2
H11	0C2	H90	OHT	HD7	MB3
H12	0C3	HA0	OPT	HD8	MB4
H20	0V1	HA1	OP1	HD9	MB5
H21	0V2	HA2	OP2	HDA	MB6
H22	0V3	HA3	OP3	HDB	MB7
H30	THT	HB0	PE	HF1	E. 1
H31	THM	HB1	PUE	HF2	E. 2
H40	FIN	HB2	RET	HF3	E. 3
H50	IPF	HC1	CTE	HF6	E. 6
H51	UVT	HC2	P24	HF7	E. 7
H60	OLT	HD1	OSD		
H70	BE	HD2	ECT		

Number of digits, 2 or 4, is automatically determined by the PLC, based on the instruction code.

The following example writes the Run Command parameters to Inverter Station #1.



No.	Item		Instruction Code	Description	Number of Data Digits
3	Run command		HFA	b7 b0: 0 0 0 0 0 0 1 0 (For example 1) [Example 1] H02 ... Forward rotation [Example 2] H00 ... Stop b0: — b1: Forward rotation (STF) b2: Reverse rotation (STR) b3: — b4: — b5: — b6: — b7: —	2 digits
4	Inverter status monitor		H7A	b7: 0 0 0 0 0 0 1 0 (For example 1) [Example 1] H02 ... During forward rotation [Example 2] H80 ... Stop due to alarm b0: Inverter running (RUN) * b1: Forward rotation (STF) b2: Reverse rotation (STR) b3: Up to frequency (SU) * b4: Overload (OL) * b5: Instantaneous power failure (IPF) * b6: Frequency detection (FU) * b7: Alarm occurrence * *The output data depends on the Pr. 190 to Pr. 195 settings.	2 digits
5	Set frequency read (E^2PROM)		H6E	Reads the set frequency (RAM or E^2PROM). H0000 to H9C40: 0.01Hz increments (hexadecimal)	4 digits
	Set frequency read (RAM)		H6D		
	Set frequency write (E^2PROM)		HEE	H0000 to H9C40: 0.01Hz increments (hexadecimal) (0 ⇒ 400.00 Hz) To change the set frequency consecutively, write data to the inverter RAM. (Instruction code: HED)	4 digits
	Set frequency write (RAM)		HED		
6	Inverter reset		HFD	H9696: Resets the inverter. As the inverter is reset on start of communication by the computer, the inverter cannot send reply data back to the computer.	4 digits
7	Alarm definition batch clear		HF4	H9696: Batch clear of alarm history	4 digits
8	All clear		HFC	All parameters return to the factory settings. Any of four different clear operations is performed according to the data. H9696: ○ × ○ ○ H9966: ○ ○ ○ ○ H5A5A: × × ○ ○ H55AA: × ○ ○ ○ When all parameter clear is executed for H9696 or H9966, communication-related parameter settings also return to the factory settings. When resuming operation, set the parameters again. *Pr. 75 value is not cleared.	4 digits
9	User clear		HFC	H9669: User clear is made. Communication Pr. ○ / Calibration × / Other Pr.* ○ / HEC HF3 HFF ○ *Pr. 75, Pr. 180 to Pr. 186 and Pr. 190 to Pr. 195 values are not cleared.	4 digits
10	Parameter write		H80 to HE3	Refer to the data list (Appendix 1) and write and/or read parameter values as required.	4 digits
11	Parameter read		H00 to H63		
12	Link parameter expansion setting	Read	H7F	H00 to H6C and H80 to HEC parameter values are changed. H00: Pr. 0 to Pr. 96 values are accessible. H01: Pr. 100 to Pr. 158 , Pr. 200 to Pr. 231 and Pr. 900 to Pr. 905 values are accessible. H02: Pr. 160 to Pr. 199 and Pr. 232 to Pr. 287 values are accessible. H03: Pr. 300 to Pr. 342 values are accessible. When using the inboard options. (When plug-in option is fitted) H09: Pr. 990 and Pr. 991 values are accessible.	2 digits
		Write	HFF		
13	Second parameter changing (Code FF = 1)	Read	H6C	When setting the programmed operation (data code H3D to H5A, HBD to HDA) parameter H00: Time H01: Time H02: Rotation direction 6 3 3 B (Time (Minute) (Second)) When setting the bias/gain (data code H5E to H6A, HDE to HED) parameter H00: Offset/gain H01: Analog H02: Analog value of terminal	2 digits
		Write	HEC		



For 2 seconds after the reset instruction is transmitted, the inverter does not accept communication.
As this period of time is controlled by the PLC, the sequence does not need to be taken into consideration

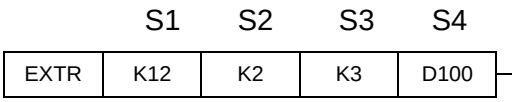
Use EXTR K12/K13 to read and write parameters

Before parameter read/ write, link parameters are automatically re-written

If a second parameter is present, it is automatically re-written

Note:
As parameters 9,10 11 and 12 are used for EXTR K12/13, DO NOT use them with EXTR K11/ K12.

5.16.1.1.7 EXTR K12 - Parameter read (Inverter to PLC)

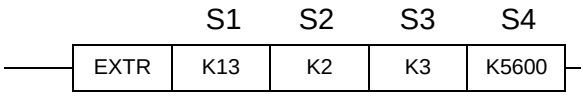


16 Bit Operation	✓
32 Bit Operation	×
Pulse - P	×

Parameter	Device type	Parameter range
S1	K/H	K12: function No. to read inverter parameters
S2	KH, D	Inverter station number (0 to 31)
S3	KH, D	Inverter instructuion code (varies with model)
S4	D, KnY, KnM, KnS	Value of storage destination

Link parameters are automatically re-written in accordance with the parameter No.

5.16.1.1.8 EXTR K13 - Parameter write (PLC to Inverter)



16 Bit Operation	✓
32 Bit Operation	×
Pulse - P	×

Parameter	Device type	Parameter range
S1	K/H	K13: function No. to write inverter parameters
S2	KH, D	Inverter station number (0 to 31)
S3	KH, D	Inverter instructuion code (varies with model)
S4	KH, D, KnX, KnY, KnM, KnS	Value to be written to inverter

Link parameters are automatically re-written in accordance with the parameter No.

5.16.1.1.9 Relationship between EXTR K12/K13 and A500/E500/S500 series

The following page is from the Data COD list of the FR A500 series instruction manual, IB(NA)-66790-G.

EXTR	K12	K(Station No.)	K(Parameter No.)	D****

The following example displays how to read the parameter from the inverter.

EXTR	K13	K(station No.)	K(Parameter No.)	K****

The following example displays how to write the parameter to the inverter.

EXTR	K12	K2	K3	D100

The following example reads the Base Frequency from the 2nd inverter station and stores this value to D100.

EXTR	K13	K2	K3	K5600

The following example writes K5600 to the Base Frequency in the 2nd inverter station.

Function	Parameter Number	Name	Data Codes		
			Read	Write	Link Parameter Extension Setting (Data code 7F/FF)
Basic functions	0	Torque boost	00	80	0
	1	Maximum frequency	01	81	0
	2	Minimum frequency	02	82	0
	3	Base frequency	03	83	0
	4	Multi-speed setting (high speed)	04	84	0
	5	Multi-speed setting (middle speed)	05	85	0
	6	Multi-speed setting (low speed)	06	86	0
	7	Acceleration time	07	87	0
	8	Deceleration time	08	88	0
	9	Electronic thermal O/L relay	09	89	0
Standard operation functions	10	DC injection brake operation frequency	0A	8A	0
	11	DC injection brake operation time	0B	8B	0
	12	DC injection brake voltage	0C	8C	0
	13	Starting frequency	0D	8D	0
	14	Load pattern selection	0E	8E	0
	15	Jog frequency	0F	8F	0
	16	Jog acceleration/deceleration time	10	90	0
	17	MRS input selection	11	91	0
	18	High-speed maximum frequency	12	92	0
	19	Base frequency voltage	13	93	0
	20	Acceleration/deceleration reference frequency	14	94	0
	21	Acceleration/deceleration time increments	15	95	0
	22	Stall prevention operation level	16	96	0
	23	Stall prevention operation level compensation factor at double speed	17	97	0
	24	Multi-speed setting (speed 4)	18	98	0
	25	Multi-speed setting (speed 5)	19	99	0
	26	Multi-speed setting (speed 6)	1A	9A	0
	27	Multi-speed setting (speed 7)	1B	9B	0
	28	Multi-speed input compensation	1C	9C	0
	29	Acceleration/deceleration pattern	1D	9D	0
	30	Regenerative function selection	1E	9E	0
	31	Frequency jump 1A	1F	9F	0
	32	Frequency jump 1B	20	A0	0
	33	Frequency jump 2A	21	A1	0
	34	Frequency jump 2B	22	A2	0
	35	Frequency jump 3A	23	A3	0
	36	Frequency jump 3B	24	A4	0
	37	Speed display	25	A5	0
Output terminal functions	41	Up-to-frequency sensitivity	29	A9	0
	42	Output frequency detection	2A	AA	0
	43	Output frequency detection for reverse rotation	2B	AB	0
Second functions	44	Second acceleration/deceleration time	2C	AC	0
	45	Second deceleration time	2D	AD	0
	46	Second torque boost	2E	AE	0
	47	Second V/F (base frequency)	2F	AF	0
	48	Second stall prevention operation current	30	B0	0
	49	Second stall prevention operation frequency	31	B1	0
	50	Second output frequency detection	32	B2	0
Display functions	52	DU/PU main display data selection	34	B4	0
	53	PU level display data selection	35	B5	0
	54	FM terminal function selection	36	B6	0
	55	Frequency monitoring reference	37	B7	0
	56	Current monitoring reference	38	B8	0
Rated output current	57	Restart coasting time	39	B9	0
	58	Restart cushion time	3A	BA	0

The data is calculated by the PLC in accordance with the parameter.

No. Reading and Writing to parameters which require a second parameter

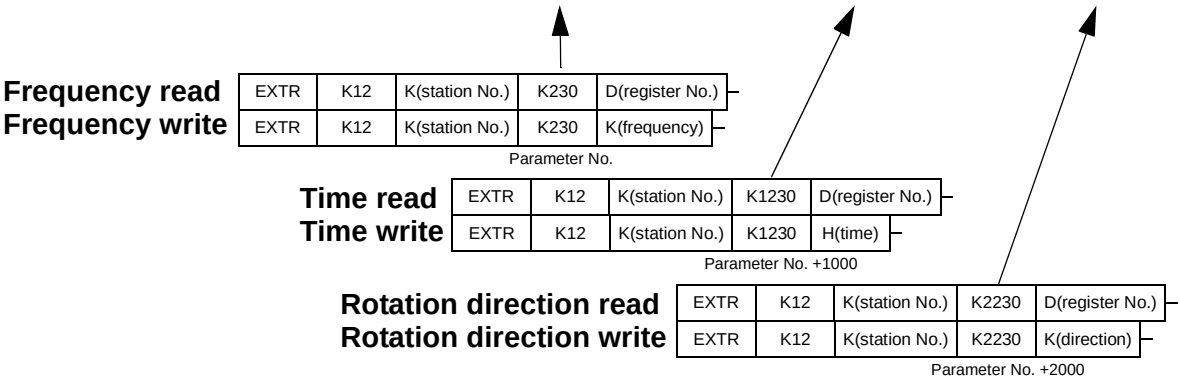
No.	Item		Instruction code	Description	Number of data digits
13	Second parameter changing (code FF=1)	Read	H6C	When setting the programmed operation (data code H3D to H5A, HBD to HDA) parameter H000: time H001: time H002: rotation direction When setting the bias/gain (data code H5E to H6A, HDE to HED) parameter H000: Offset/Gain H001: Analog H002: Analog value of terminal	2 digits
		Write	HEC		

For parameters that require the setting of a second parameter. If a value of '+0', '+1000' or '+2000' is set to a parameter No. in the inverter manual, the second parameter will be automatically re-written before a general parameter read/write.

	Classification	Offset for parameter number
a)	H00: Operation frequency	+0
	H01: Time	+1000
	H02: Rotation direction	+2000
b)	H00: Offset / Gain	+0
	H01: Analog	+1000
	H02: Analog value of terminal	+2000

Setting the third parameter of EXTR K12/K13 during programmed operation in the A500

Parameter No.	Name	Operation frequency read/write	Time read/write	Rotation direction read/write
201	Program set 1	201	1201	2201
202	Program set 1	202	1202	2202
203	Program set 1	203	1203	2203
204	Program set 1	204	1204	2204
205	Program set 1	205	1205	2205
206	Program set 1	206	1206	2206
207	Program set 1	207	1207	2207
208	Program set 1	208	1208	2208
209	Program set 1	209	1209	2209
210	Program set 1	210	1210	2210
211	Program set 2	211	1211	2211
212	Program set 2	212	1212	2212
213	Program set 2	213	1213	2213
214	Program set 2	214	1214	2214
215	Program set 2	215	1215	2215
216	Program set 2	216	1216	2216
217	Program set 2	217	1217	2217
218	Program set 2	218	1218	2218
219	Program set 2	219	1219	2219
220	Program set 2	220	1220	2220
221	Program set 3	221	1221	2221
222	Program set 3	222	1222	2222
223	Program set 3	223	1223	2223
224	Program set 3	224	1224	2224
225	Program set 3	225	1225	2225
226	Program set 3	226	1226	2226
227	Program set 3	227	1227	2227
228	Program set 3	228	1228	2228
229	Program set 3	229	1229	2229
230	Program set 3	230	1230	2230



Reading and writing the bias/gain in the A500/E500/S500

Parameter No.	Name	Offset/gain read/write	Analog read/write	Terminal analog value read
902	Frequency setting voltage bias	902	1902	2902
903	Frequency setting voltage gain	903	1903	2903
904	Frequency setting current bias	904	1904	2904
905	Frequency setting current gain	905	1905	2905

Offset/Gain read

EXTR	K12	K(station No.)	K902	D(register No.)

Offset/Gain write

EXTR	K12	K(station No.)	K902	K(frequency)

Parameter No.

Analog read

EXTR	K12	K(station No.)	K1902	D(register No.)

Analog write

EXTR	K12	K(station No.)	K1902	K(percentage)

Parameter No.+1000

Terminal analog value read

EXTR	K12	K(station No.)	K2902	D100

Parameter No.+2000

The following parameters CANNOT be used with EXTR K12/K13

Function	Parameter No.	Name		Data code		
				Read	Write	Link parameter extension set value (data code 7F/FF)
	-	Second parameter changing		6C	EC	-
	-	Frequency setting	Operation frequency (RAM)	6D	ED	-
	-		Operation frequency (E^2PROM)	6E	EE	-
	-	Monitoring	Frequency monitoring	6F	-	-
	-		Output current monitoring	70	-	-
	-		Output voltage monitoring	71	-	-
	-		Special monitoring	72	-	-
	-		Special monitoring selection No.	73	F3	-
	-	Alarm display	Most recent No.1, No.2 / alarm display clear	74	F4	-
	-		Most recent No.3, No.4	75	-	-
	-		Most recent No.5, No.6	76	-	-
	-		Most recent No.7, No.8	77	-	-
	-	Inverter status monitoring / run command		7A	FA	-
	-	Operation mode aquisition		7B	FB	-
	-	Parameter all clear		-	FC	-
	-	Inverter reset		-	FD	-
	-	Link parameter extension setting		7F	FF	-

Note:
Parameters 77 and 79 are accessible for computer link operation, but they are NOT available if a FR-A5NR is used as they need the PU connector.

- Definition of special D registers and special M coils

M8154	Offers debugging function.	D8154	Waiting time for a response from the inverter.
M8155	ON during communication, and OFF when communication is complete.	D8155	Step No. of the instruction which is executing a function related to the inverter. Stores '-1' while communication is not executed
M8156	Turns ON when a communication error occurs. Effective only just after the EXTR instruction is executed. When the next EXTR instruction is executed, M8156 is cleared.	D8156	Communication command error code. Updated when an error occurs in the EXTR instruction at the next time. Initialized to '-1' by STOP ⇨ RUN.
M8157	Turns ON when a communication error occurs (latch). Reset by STOP ⇨ RUN	D8157	Step No. in which a communication command error has occurred' (Latches the step No. which occurred for the first time after start of run.) Initialized to '-1' by STOP ⇨ RUN

- D8154: Waiting time for response from the inverter
  If the inverter does not give a response within the time set here, after the PLC has transmitted a command, it is regarded as no response.
  When '0' is set to D8154, if the inverter does not give a response in 2 seconds, it is processed as an error.
  The value is set to D8154, multiplied by 0.1(s) is treated as the judgement time for no response.

- Use EXTR K10 (INV MON), K11 (INV CMD), K12 (RD PARAM) and K13 (WR PARAM) in accordance with the contents of read/write communication to/from the inverter.

- For EXTR K12/K13, the PLC automatically re-writes the link parameters in accordance with the parameter No.
  For parameters relating to a second parameter of the inverter, program them using the parameter No. adding by '+0', '+1,000' or '+2,000'.

- EXTR K10 to EXTR K13 repeatedly execute communication while the drive condition is ON.

- If two or more read instructions are driven at the same time, when the first is completed, next is automatically executed. The step No. being executed is stored in D8155.

- Communication start
  If communication is driven while the comms port is open, communication starts.
  If the drive condition turns OFF during communication, communication continues until it is completed. (The system will be adversley affected if communication is aborted by turning OFF the drive condition.)

- Debugging function by M8154
  A standby time of 15 ms is assured after communication with the inverter is completed until next communication starts. While M8154 is ON, the standby time becomes 1,000 ms. By monitoring D8156, the user can confirm the step which is executing communication.

5.16.1.1.10 Consistency with other instructions

- STL instruction
  During communication, if the executed state is set to OFF, the communication port is not open. As a result, communication is disabled.

- Branch instructions CJ and CJP
  During communication, if the EXTR instruction is skipped by a CJ or CJP instruction, the communication port is not open. As a result, communication is disabled.

- Description in subroutine
  As the EXTR instruction requires the time of two or more operation cycles until execution is complete, it is prohibited to write a subroutine where the EXTR instruction is called twice or more in one operation cycle.

- Inside master control
  No problems are expected.

- FOR-NEXT
  It is prohibited to use an EXTR instruction together with a FOR-NEXT instruction.

- Description in interrupt
  It is prohibited to describe an EXTR instruction in any interrupt.

- Cautions on write during run
  (1) It is prohibited to rewrite the function No. of theEXTR instructions first parameter.
  (If the function No. is rewritten during run, a problem will occur in the same way as change in the application instruction No.)
  (2) It is prohibited to delete an EXTR instruction.
  (If the EXTR instruction is deleted during run, communication will be disabled.)

- Communication complete
  When communication is finished, the completion flag M8029 turns ON, without regard to the completion status (normal or abnormal).
  (M8029 turns ON for one calculation cycle at the time of completion.
  M8029 is used by manu instructions and therefore the ON/OFF status of M8029 is held only until the next instruction which utilizes M8029 is executed.)

- Communication error
  Communication is executed three times in total, including two retries. If communication is abnormally finished even after the third execution, it is regarded as an error. Error types are classified as follows.

1) When an error code is returned from the inverter
2) When the inverter does not give any response
3) When a response is given by an unspecified station
4) When a receive error (such as overrun, parity error and framing error) occurs
5) When M8063 turns ON and error code 6301 is set to D8067
6) When the check sum of the data returned by the inverter does not match

For 1), 2) & 3) M8156 is set to ON, and an error code is set to D8156.
If a communication error occurs, it is cleared when the next EXTR K10/K11/K12/K13 is executed.

In general when an error occurs, M8157 turns ON and remains ON (latch) until it is set OFF.

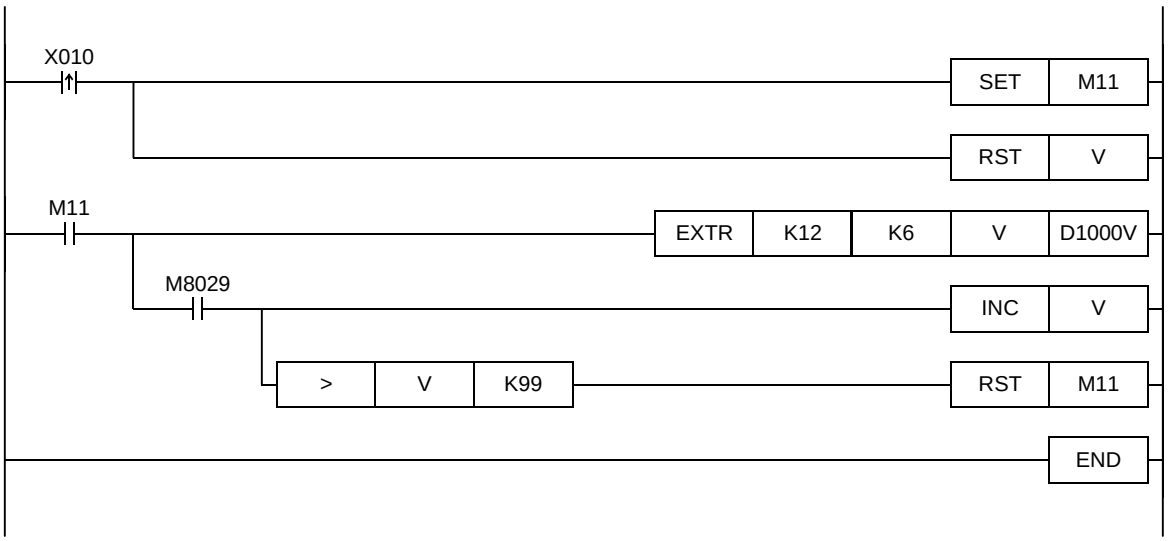
5.16.1.1.11 Communication command error codes

The table below shows values set to D8156 after EXTR K10 K13 are executed.

D8155	Contents of error	Inverter operation
H0000	Communication is terminated normally (no error)	
H0001	The inverter does not give any response	
H0002	Timeout error interlocking with M8129. Error occurs when transmission from the inverter is aborted	
H0003	An unspecified station has given a response	
H0004	The sum of the data returned by the inverter does not match.	
H0005	In parameter read/write, parameters Nos. 400 to 899 are specified but cannot be supported. Sets error code 6702 into D8067.	
H006	The communication port is being used for another function and therefore cannot be used for the EXTR instruction. Sets error code 6702 into D8067.	
H0100	The inverter has transmitted the error code H0 - Computer NAK error. Communication request data includes an error beyond the permissible number of retries	If an error occurs beyond the permissible number of retries, the inverter will come to an alarm stop
H0101	The inverter has transmitted the error code H1 - Parity error. The contents are different from the specified parity	
H0102	The inverter has transmitted the error code H2 - Sum check error. The sum check code value in the computer is different to that of the inverter	
H0103	The inverter has transmitted the error code H3 - Protocol error. There is a grammar error in the data received by the inverter, data receive is not completed within the specified time, or the CR or LF is different from the parameter setting	
H0104	The inverter has transmitted the error code H4 - Framing error. The stop bit length is different from the default set value	
H0105	The inverter has transmitted the error code H5 - Overrun error. Data sent before previous receive transmission was complete.	
H0106	The inverter has transmitted the error code H6. Currently undefined	
H0107	The inverter has transmitted the error code H7 - Character error. An unused character (any character other than 0 to 9, A to F and control codes) is received	The inverter does not accept the data nor does it come to an alarm stop.
H0108	The inverter has transmitted the error code H8. Currently undefined	
H0109	The inverter has transmitted the error code H9. Currently undefined	
H010A	The inverter has transmitted the error code HA. This is a mode error. A parameter write was tried while the inverter was in operation or computer link mode was not selected.	The inverter does not accept the data nor does it come to an alarm stop.
H010B	The inverter has transmitted the error code HB -Instruction code error. A non-existing instruction code is specified	
H010C	The inverter has transmitted the error code HC -Data range error. In a parameter write, data outside the permissible setting range is specified. The inverter does not accept the data and an alarm does not occur	
H010D	The inverter has transmitted the error code HD. Currently undefined	
H010E	The inverter has transmitted the error code HE. Currently undefined	
H010F	The inverter has transmitted the error code HF. Currently undefined	

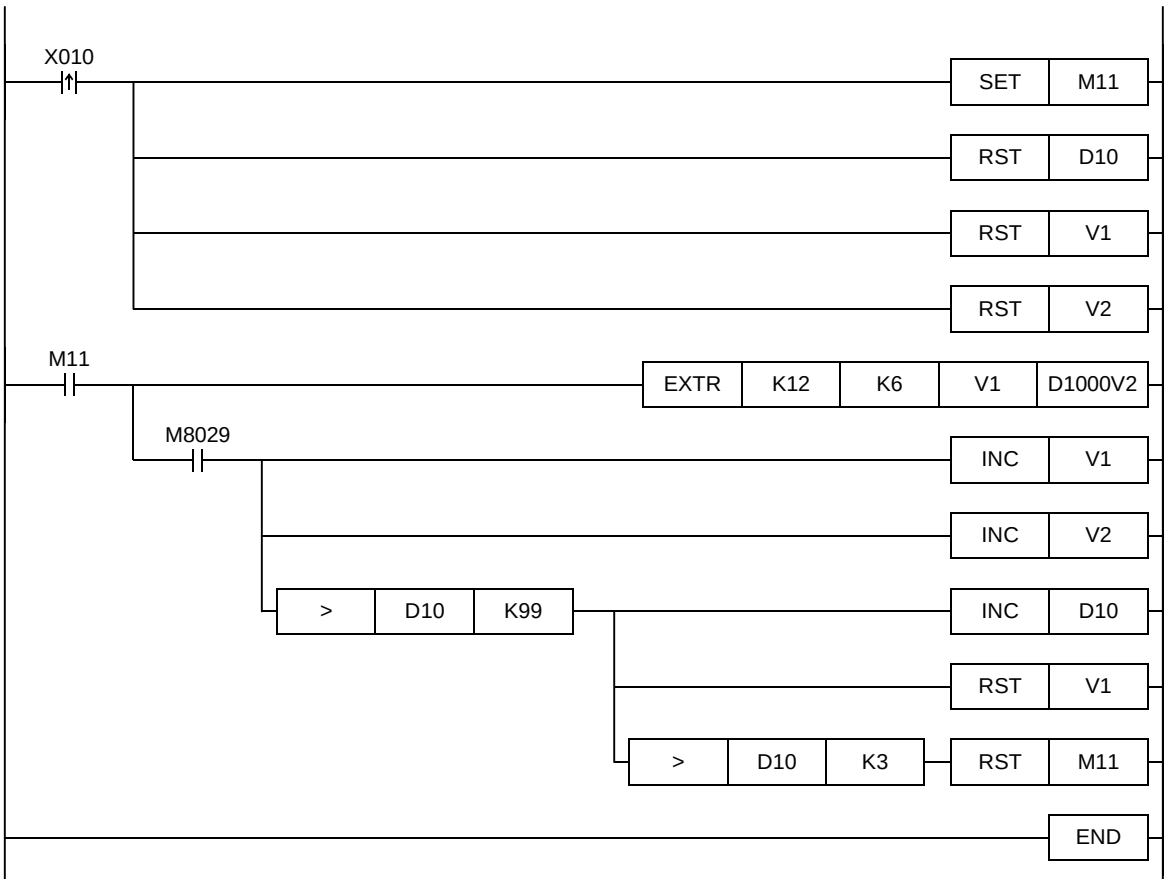
5.16.1.1.12 Example program 1

This program reads parameters 0 to 99 in the inverter at station No. 6, to D1000 to D1099 in the PLC.



5.16.1.1.13 Example program 2

This program reads parameters 0 to 99 in the inverters at station Nos. 6, 7, 8 and 9, to D1000 to D1099, D1100 to D1199, D1200 to D1299 and D1300 to D1399 respectively in the PLC.



V Station No. control

V1 Parameter No. control

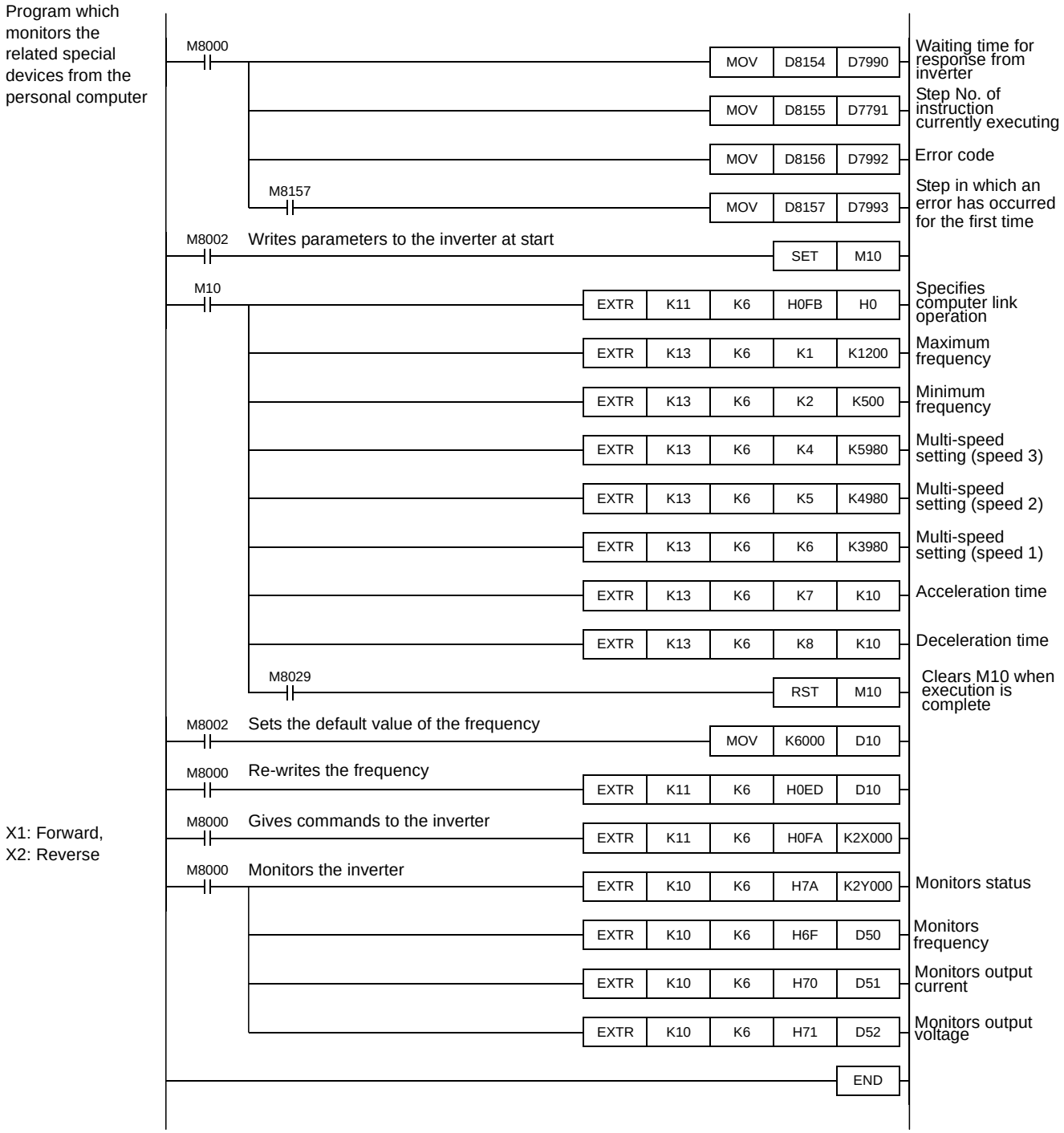
V2 Read parameter storage destination

5.16.1.1.14 Example program 3

This program writes the speed parameter from PLC to inverter, performs forward rotation by input X1, and reverse rotation by input X2.

By re-writing D10 in the peripheral equipment or the display unit, the frequency of the inverter can be changed.

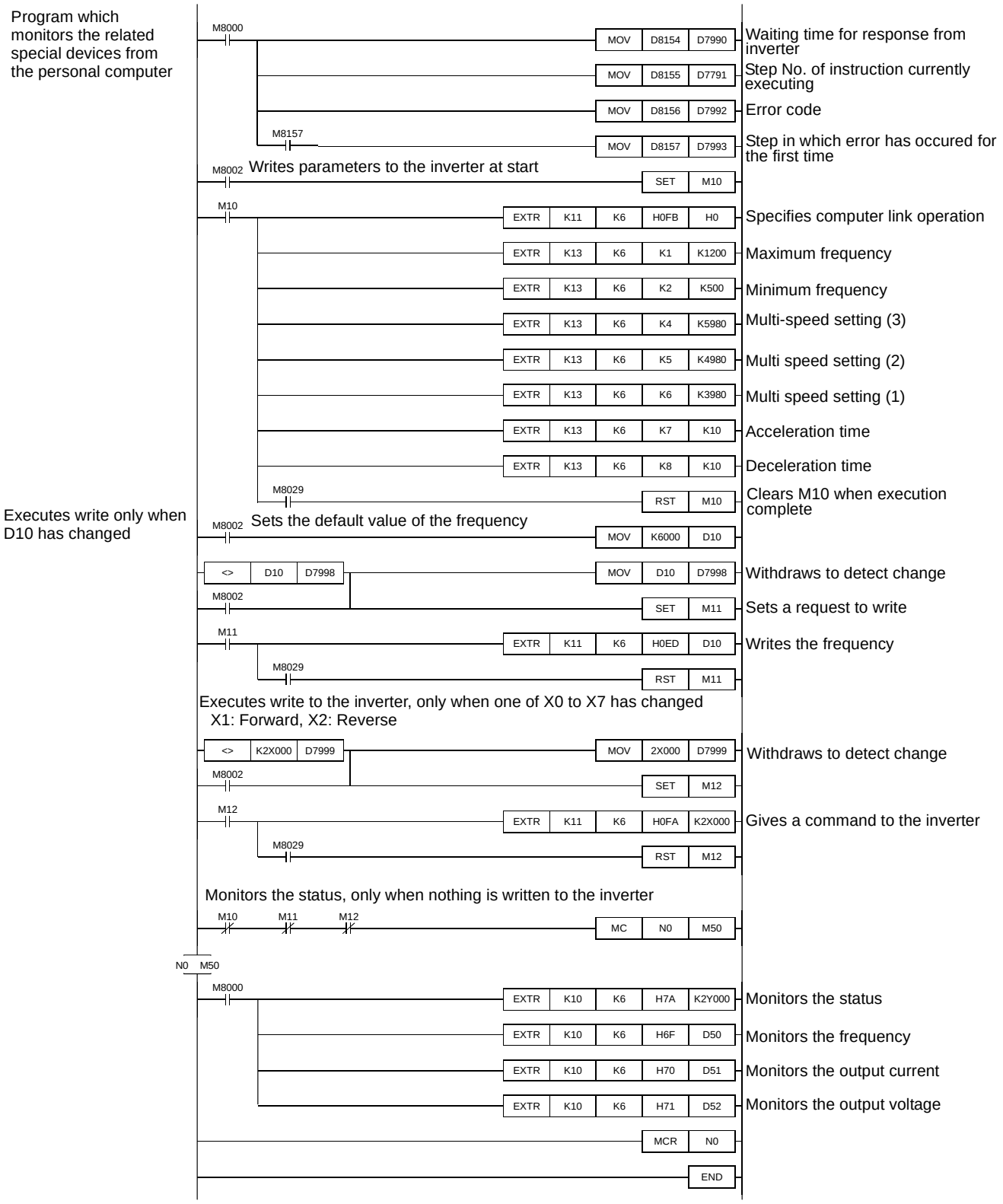
This program also monitors the frequency and output current in the inverter.



5.16.1.1.15 Example program 4

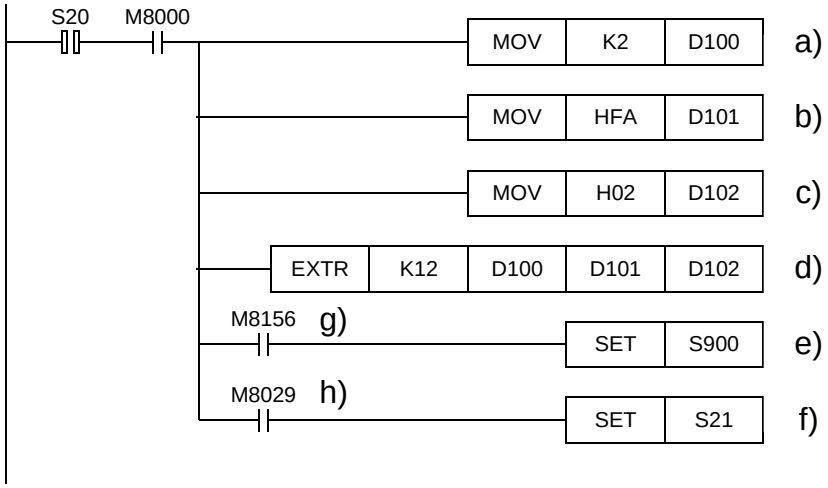
In the previous example, monitoring and write to the inverter are always driven. If the program changes the frequency or gives a forward/reverse rotation command, communication with the inverter may be delayed depending on the step executing communication.

In the example below, when a request to write is generated, a request to read is interrupted, write is executed, then monitoring is continued again after write is completed.



5.16.1.1.16 Example program 5

Example using the STL instruction



a) Specifies station No. 2

b) Instruction code for operation command

c) Forward rotation command

d) Transmits/receives a command to/from the inverter.

e) Changes to the 'error processing' state as an error has occurred.

f) Changes to the 'next' state as receive is normally finished.

g) Receive is abnormally finished.

h) Receive is complete.

5.16.1.1.17 Related Error Code Lists

PLC hardware error code list (M8061, D8061)

K6110	There is an abnormality in the extension ROM cassette.

Grammar error code list (M8065, D8065)

K6512	The FNC 180 is described while the extension ROM cassette is not mounted.

Operation error code list (M8067, D8067)

K6760	The sum of the value read by the ABS instruction does not match.
K6761	When FNC185 was executed, the extension memory cassette was not mounted. Or, the firmware of the function No. specified by the first parameter does not exist in the extension memory cassette.

MEMO

Applied Instructions:

FX1S	FX1N	FX2N	FX2NC

5.17 Inline Comparisons - FNC 220 to FNC 249

Contents:

Symbols list:

D - Destination device.
S - Source device.
m, n- Number of active devices, bits or an operational constant.
Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D_1, S_3 or for lists/tabled devices D_{3+0}, S_{+9} etc.
MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.
LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆ - A 32 bit mode instruction modified to use pulse (single) operation.

✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

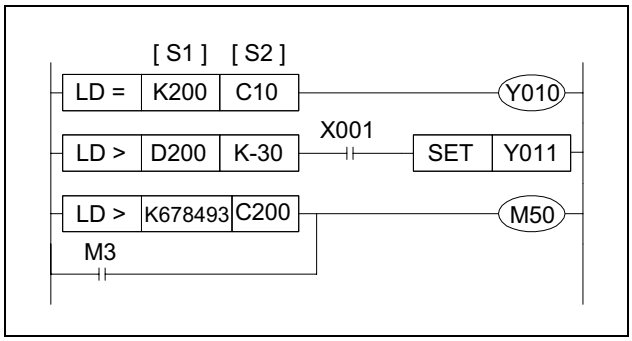
✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.17.1 LD compare (FNC 224 to 230)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
LD❒ (LoaD compare) where ❒ is =, >, <, <>, ≤, ≥	Initial comparison contact. Active when the comparison S1 ❒ S2 is true.	K,H, KnX, KnY, KnM, KnS, T, C, D, V, Z	K,H, KnX, KnY, KnM, KnS, T, C, D, V, Z	LD❒: 5 steps DLD❒: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The value of S1 and S2 are tested according to the comparison of the instruction. If the comparison is true then the LD contact is active. If the comparison is false then the LD contact is not active.

Points to note:

The LD comparison functions can be placed anywhere in a program that a standard LD instruction can be placed. I.e., it always starts a new block.
(See page 2-3 for LD instruction)

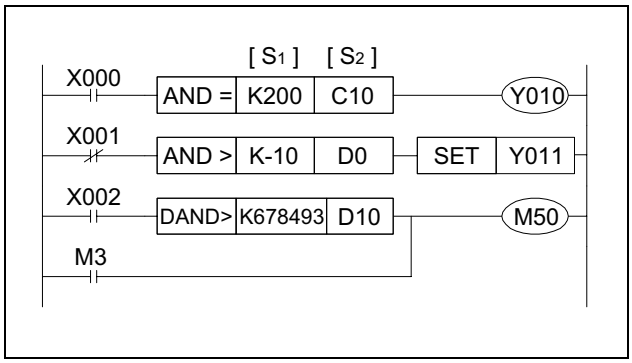
FNC No.	Mnemonic		Active when	Inactive when
	16 bit	32 bit		
224	LD =	DLD =	S1 = S2	S1 ≠ S2
225	LD >	DLD >	S1 > S2	S1 ≤ S2
226	LD <	DLD <	S1 < S2	S1 ≥ S2
228	LD <>	DLD <>	S1 ≠ S2	S1 = S2
229	LD ≤	DLD ≤	S1 ≤ S2	S1 > S2
230	LD ≥	DLD ≥	S1 ≥ S2	S1 < S2

5.17.2 AND compare (FNC 232 to 238)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
AND❒ (AND compare) where ❒ is =, >, <, <>, ≤, ≥	Serial comparison contact. Active when the comparison S1 ❒ S2 is true.	K,H, KnX, KnY, KnM, KnS, T, C, D, V, Z	K,H, KnX, KnY, KnM, KnS, T, C, D, V, Z	AND❒: 5 steps DAND❒: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The value of S_1 and S_2 are tested according to the comparison of the instruction. If the comparison is true then the AND contact is active. If the comparison is false then the AND contact is not active.

Points to note:

The AND comparison functions can be placed anywhere in a program that a standard AND instruction can be placed. i.e., it is a serial connection contact.
(See page 2-6 for AND instruction)

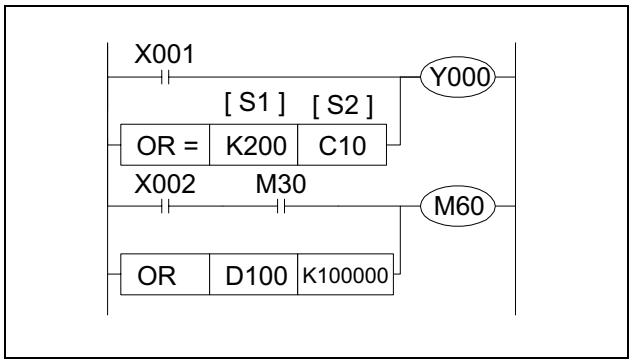
FNC No.	Mnemonic		Active when	Inactive when
	16 bit	32 bit		
232	AND =	DAND =	$S_1 = S_2$	$S_1 \neq S_2$
233	AND >	DAND >	$S_1 > S_2$	$S_1 \leq S_2$
234	AND <	DAND <	$S_1 < S_2$	$S_1 \geq S_2$
236	AND <>	DAND <>	$S_1 \neq S_2$	$S_1 = S_2$
237	AND ≤	DAND ≤	$S_1 \leq S_2$	$S_1 > S_2$
238	AND ≥	DAND ≥	$S_1 \geq S_2$	$S_1 < S_2$

5.17.3 OR compare (FNC 240 to 246)

FX1S	FX1N	FX2N	FX2NC

Mnemonic	Function	Operands		Program steps
		S	D	
OR❒ (OR compare) where ❒ is =, >, <, <>, ≤, ≥	Parallel comparison contact. Active when the comparison S1 ❒ S2 is true.	K,H, KnX, KnY, KnM, KnS, T, C, D, V, Z	K,H, KnX, KnY, KnM, KnS, T, C, D, V, Z	OR❒: 5 steps DOR❒: 9 steps

16 BIT OPERATION	32 BIT OPERATION	PULSE-P



Operation:

The value of S_1 and S_2 are tested according to the comparison of the instruction. If the comparison is true then the OR contact is active. If the comparison is false then the OR contact is not active.

Points to note:

The OR comparison functions can be placed anywhere in a program that a standard OR instruction can be placed. i.e., it is a parallel connection contact.
(See page 2-7 for OR instruction)

FNC No.	Mnemonic		Active when	Inactive when
	16 bit	32 bit		
240	OR =	DOR =	$S_1 = S_2$	$S_1 \neq S_2$
241	OR >	DOR >	$S_1 > S_2$	$S_1 \leq S_2$
242	OR <	DOR <	$S_1 < S_2$	$S_1 \geq S_2$
244	OR <>	DOR <>	$S_1 \neq S_2$	$S_1 = S_2$
245	OR ≤	DOR ≤	$S_1 \leq S_2$	$S_1 > S_2$
246	OR ≥	DOR ≥	$S_1 \geq S_2$	$S_1 < S_2$

MEMO