

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.

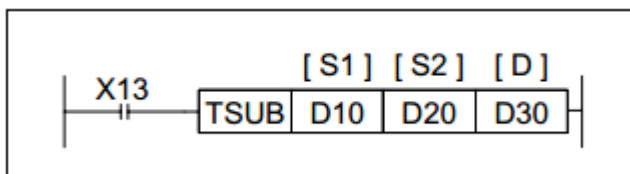
Points to note:

- 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)

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.

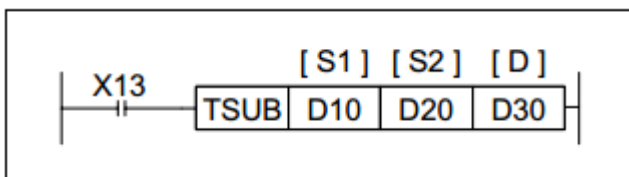
c) If the addition of the two times results in a value of zero (0:00:00: 0 hours, 0 minutes, 0 seconds) then the zero flag M8020 is set ON.

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)

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
TSUB FNC 163 (Time Subtraction)	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 S1,S2and D specify the head address of 3 data devices to

be used a time value.

The time value in S1is subtracted from the time

valueinS2, 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.

S1		S2		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.

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	16:04:55
		M8021 ON

When this happens the borrow flag M8021 is set ON.

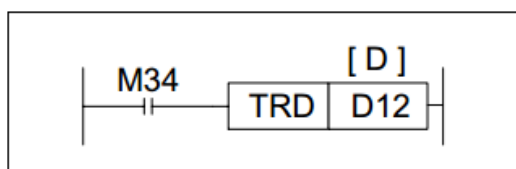
c) If the subtraction of the two times results in a value of zero (00:00:00 hours) then the zero flag M8020 is set ON.

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)

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

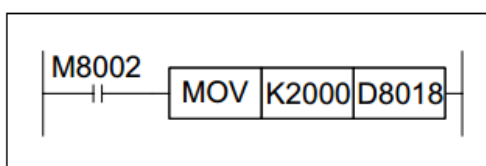
devices specified by the head address D.

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.

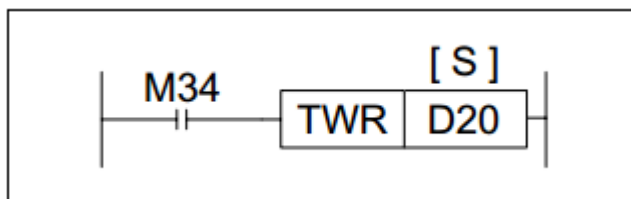


If this is done then the clock year should not be used during the first scan as it will be a two digit number before the instruction and a value of 2000 after the instruction until the END instruction executes. After the first scan the year is read and written as a 4 digit number.

5.14.6 TWR (FNC 167)

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

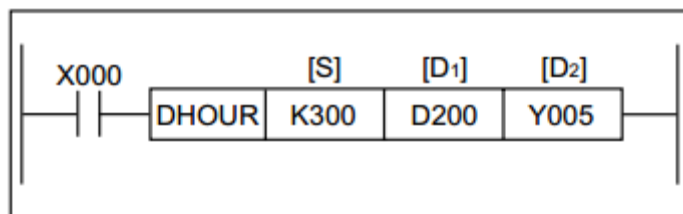
Points to note:

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)

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
------------------	------------------	---------



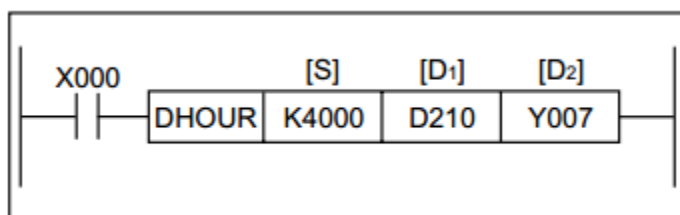
Operation 1: 16 bit

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

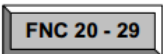
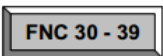
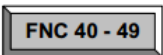
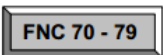
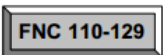
time in which [D 2] turns on (Hrs) [D 1] = Current value in Hours [D 1]+2 = Current value, if less than 1 hour. In seconds [D 2] = Alarm output destination, when [D 1] exceeds [S] In the above example, [D 2] turns on at 4000 hours and 1 second.

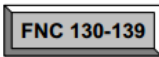
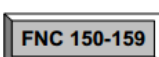
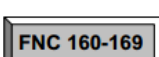
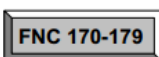
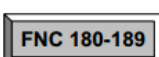
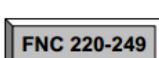
Points to note:

- 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.
- The hour meter will continue operation even after the alarm output [D2] turns ON. Operation will stop when the value of [D1] reaches the maximum for the specified 16 or 32 bit operation.

If continuous operation id required, clear the value stored in [D 1]to[D1]+1 (16-bit) and [D 1]to [D 1]+2 (32-bit).

Applied Instructions:

1.		Program Flow	5-4
2.		Move And Compare	5-16
3.		Arithmetic And Logical Operations (+, -, ×, ÷)	5-24
4.		Rotation And Shift	5-34
5.		Data Operation	5-42
6.		High Speed Processing	5-52
7.		Handy Instructions	5-66
8.		External FX I/O Devices	5-80
9.		External FX Serial Devices	5-94
10.		Floating Point 1 & 2	5-110

11.		Trigonometry (Floating Point 3)	5-118	
12.		Data Operations 2	5-122	
13.		Positioning Control	5-126	
14.		Real Time Clock Control	5-136	
	15.		Gray Codes	5-146
	16.		Additional Functions	5-146
	17.		In-line Comparisons	5-150



② 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. D1,S3or for lists/tabled devices D3+0,S+9etc.

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

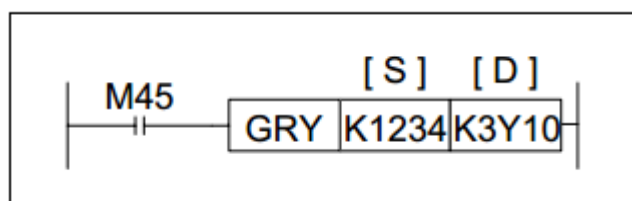
 - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

 - 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)

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.

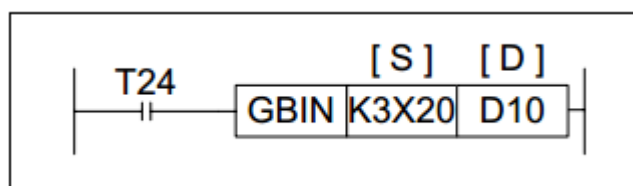
Points to Note:

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)

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:

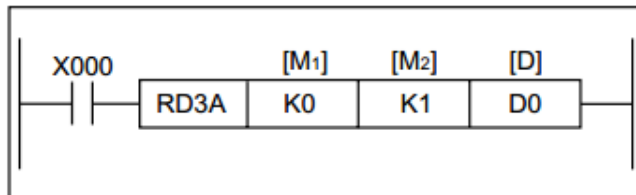
This instruction can be used to read the value from a gray code encoder.

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)

Mnemonic	Function	Operands			Program steps
		M1	M2	D	
RD3A FNC 176 Read analog block	Analog block read for TX _{0N-3A} , TX _{2N-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:

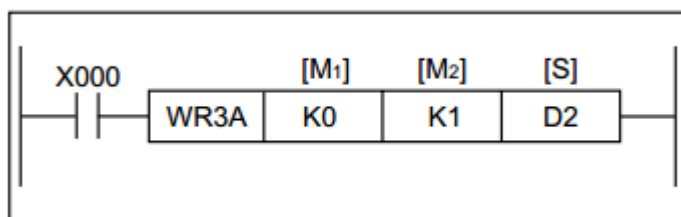
[M 1] = Special block number,
K0 to K7

[M 2] = Analog input channel number, K1/K21 or K2/K22 [D] = Read data

5.15.4 WR3A (FNC 177)

Mnemonic	Function	Operands			Program steps
		M1	M2	S	
WR3A FNC 177 Write to Analog block	Write data to analog block TX _{0N-3A} , TX _{2N-2AD} , TX _{2N-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
------------------	------------------	---------

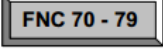
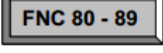


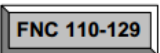
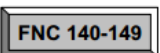
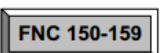
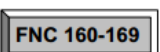
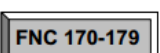
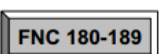
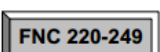
Operation:

This instruction writes data to the [M 1] = Special block number, K0 to K7

[M 2] = Analog output channel number, K1/K21 or K22 [S] = Write data

Applied Instructions:

1.		Program Flow	5-4
2.		Move And Compare	5-16
3.		Arithmetic And Logical Operations (+, -, ×, ÷)	5-24
4.		Rotation And Shift	5-34
5.		Data Operation	5-42
6.		High Speed Processing	5-52
7.		Handy Instructions	5-66
8.		External FX I/O Devices	5-80
9.		External FX Serial Devices	5-94

10.		Floating Point 1 & 2	5-110
11.		Trigonometry (Floating Point 3)	5-118
12.		Data Operations 2	5-122
13.		Positioning Control	5-126
14.		Real Time Clock Control	5-136
15.		Gray Codes	5-146
 16.		Additional Functions	5-146
17.		In-line Comparisons	5-150



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. D1,S3or for lists/abled devices D3+0,S+9etc.

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

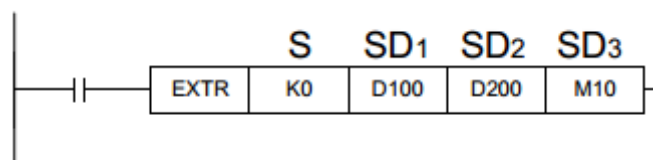
✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

✉ - 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)

Mnemonic	Function	Operands			Program steps
		S	SD ₁	SD ₂ SD ₃	
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.

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 BD is attached to the PLC, for communication with a HCFA 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.

For HC-PCS/WIN-E:

- 1) Select “Option” - “Serial setting (Parameter)”

2) Click “Yes”

3) Set “Serial setting (Parameters)” as shown below

Serial setting (parameter)

Protocol	RS instruction	a
Data bits	7	
Parity	Even	
Stop bits	1	
Transfer speed (bps)	9600	b
Header	OFF	
Terminator	OFF	
Control line	None	
Hardware	RS-485	c
Control mode	None	
Sum check	No	
Control procedure	Format 1	
Station number	00	H
Comms timeout	1	x 10ms

Click the [Clear] button to clear the setting above.
[When programming software transfer the program to the communication boards, click the [Clear] button and clear the special resistor D8120 in the PLC to "0".]

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 HCP-Works

- 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

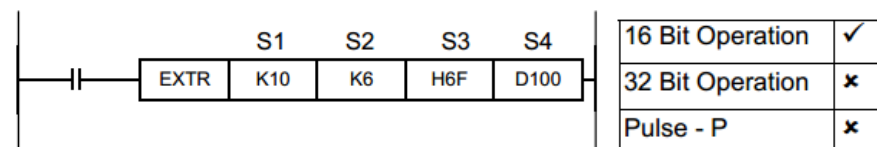
Inverter communication specification and application to PLC

		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:

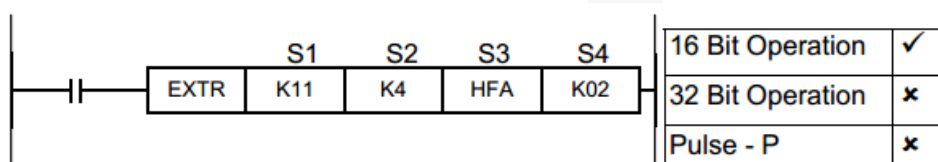
Some of the specifications above are fixed in the PLC but variable in the inverter. This has been done to ease set up and reduce any possible problems during configuration.

5.16.1.1.4 EXTR K10 - Monitoring operations (Inverter to PLC)



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

5.16.1.1.5 EXTR K11 - Control operations (PLC to Inverter)



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

5.16.1.1.6 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 the EXTR 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.

5.16.1.1.7 Communication command error codes

The table below shows values set to D8156 after EXTR K10 K13 are executed.

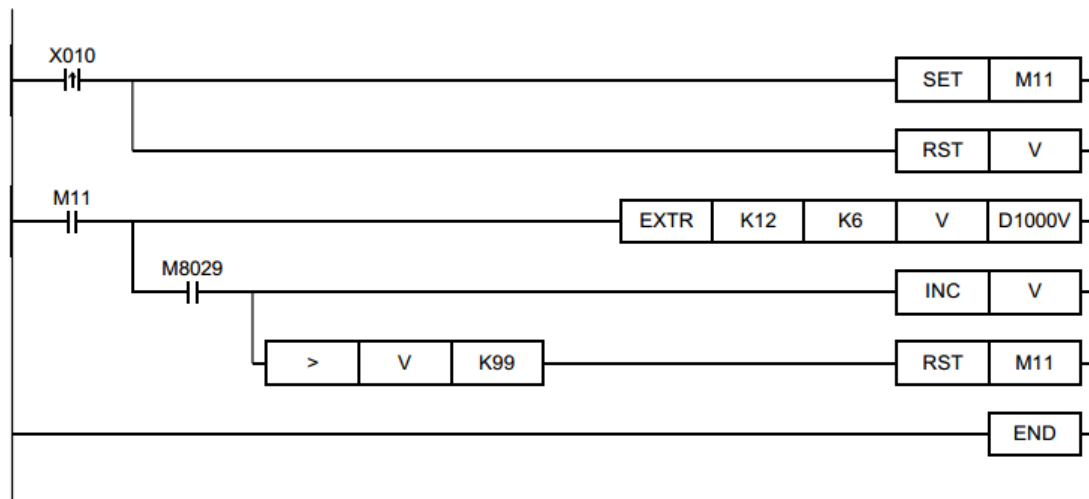
D8155	Contents of error	Inverter operation
H0000	Communication is terminated normally (no error)	
H0001	The inverter does not give any response	
H0002	Timeout error interlocking with M8129. Error occurs when transmission from the inverter is aborted	
H0003	An unspecified station has given a response	
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 st
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	

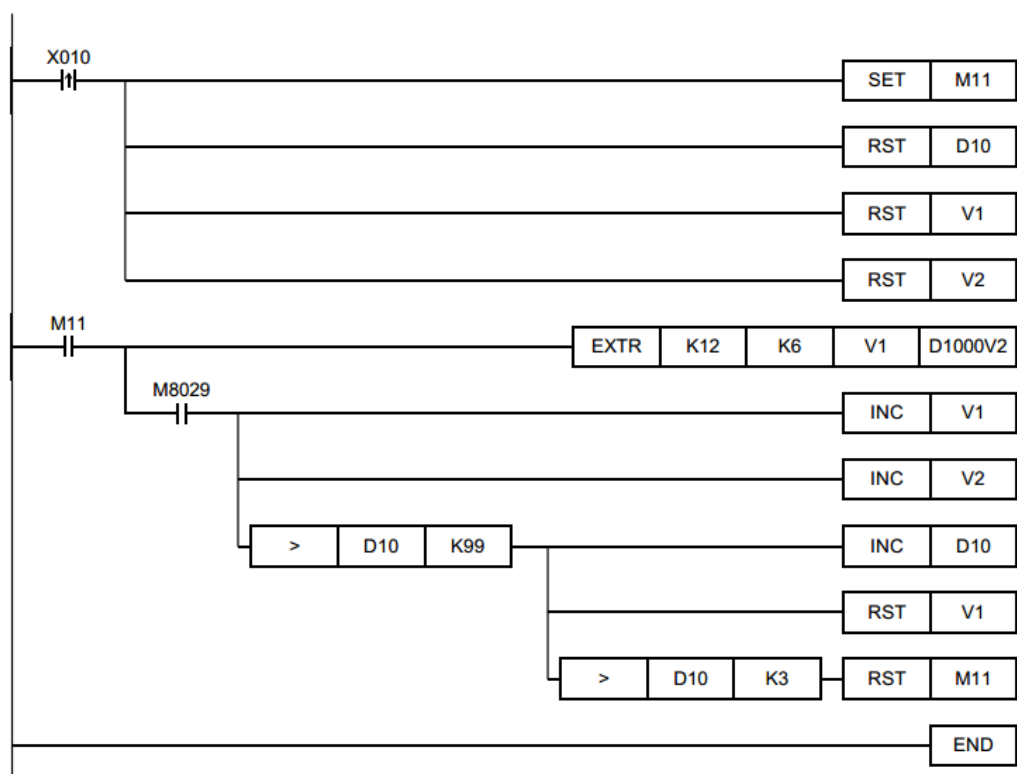
5.16.1.1.8 Example program 1

This program reads parameters 0 to 99 in the inverter at station No. 6, to D1000 to D1099 in the PLC.



5.16.1.1.9 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

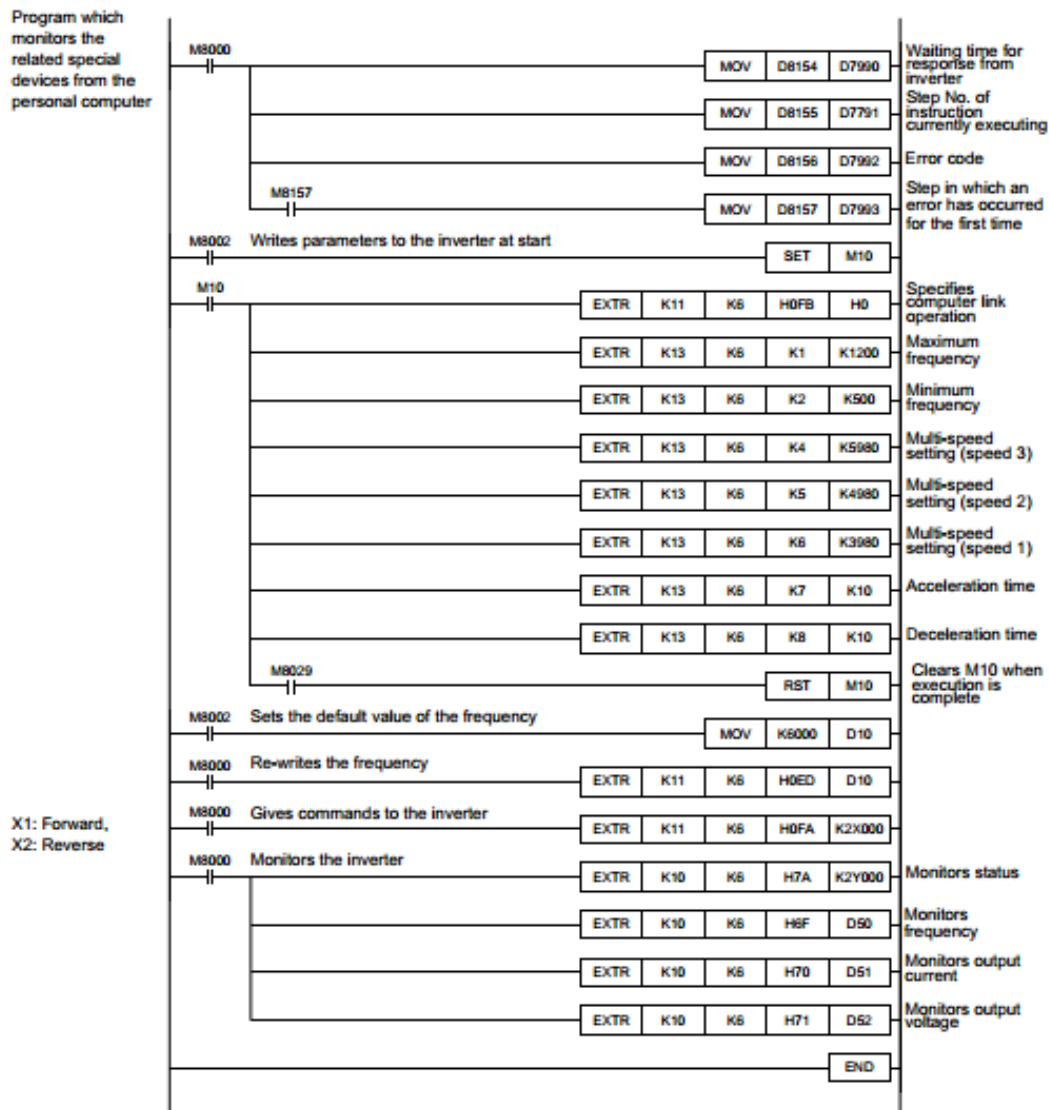
V2 Read parameter storage destination

5.16.1.1.10 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.

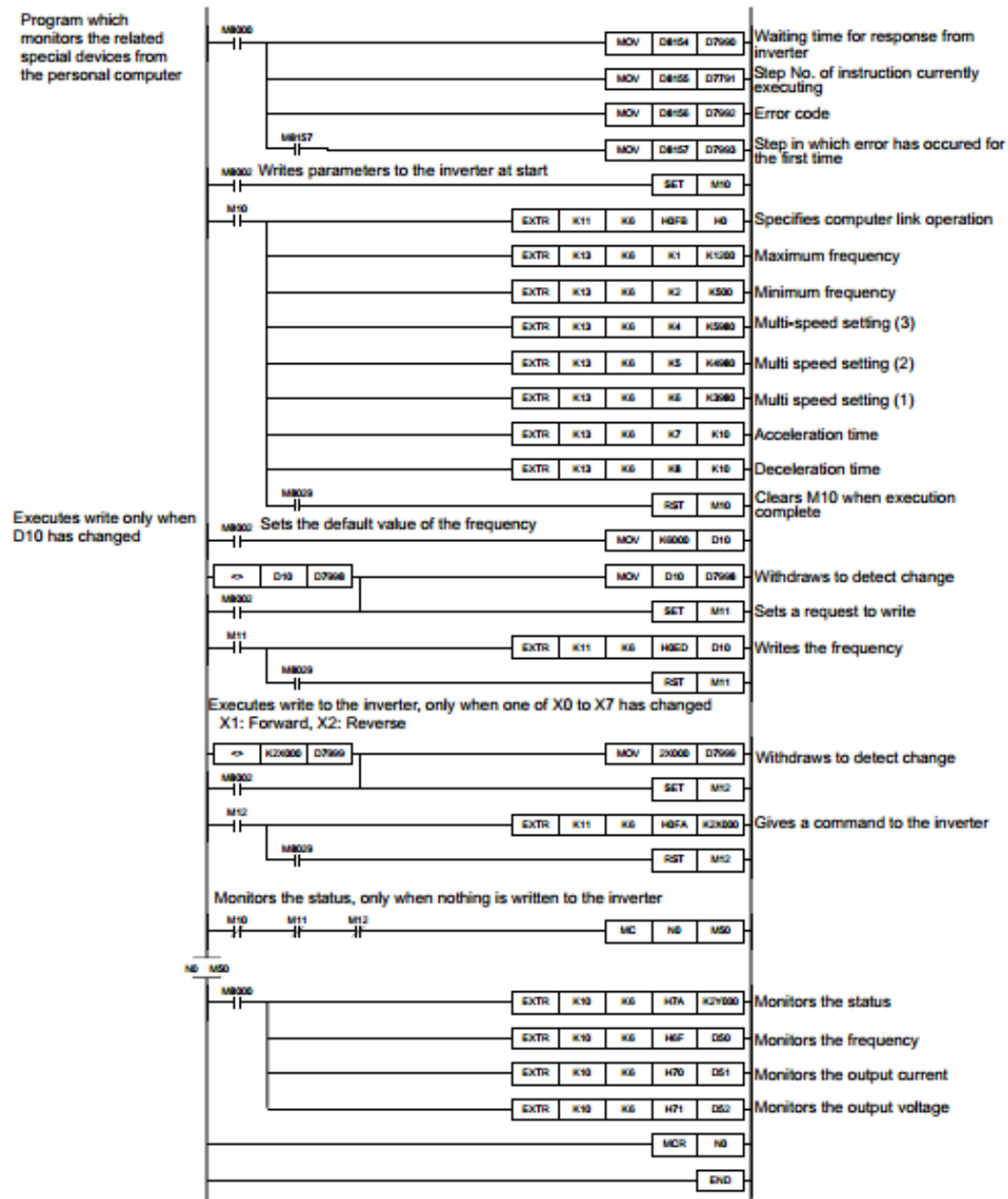
This program also monitors the frequency and output current in the inverter.



5.16.1.1.11 Example program 4

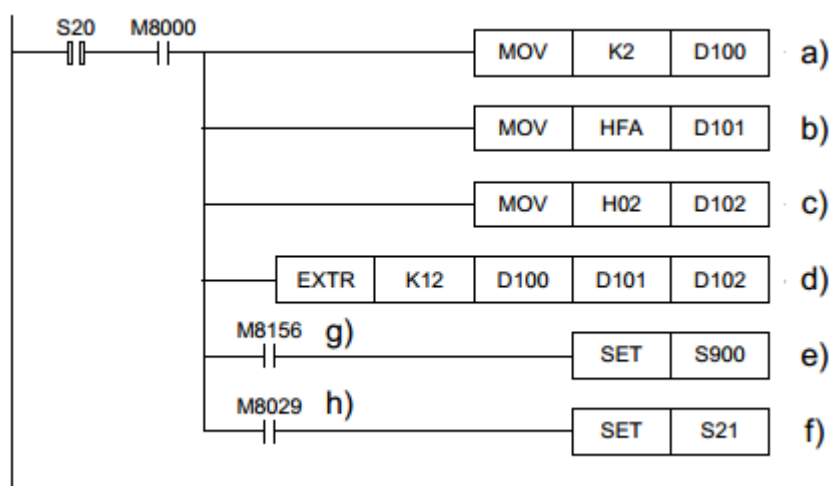
In the previous example, monitoring and write to the inverter are always driven. If the program changes the frequency or gives a forward/reverse rotation command, communication with the inverter may be delayed depending on the step executing communication.

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.



5.16.1.1.12 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.

5.16.1.1.13 Related Error Code Lists

PLC hardware error code list (M8061, D8061)

K6110	There is an abnormality in the extension ROM cassette.
--------------	---

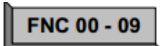
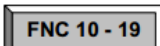
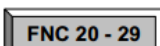
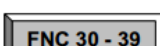
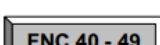
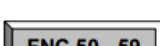
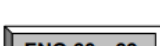
Grammar error code list (M8065, D8065)

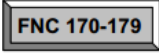
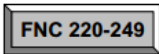
K6512	The FNC 180 is described while the extension ROM cassette is not mounted.
--------------	--

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.

Applied Instructions:

- | | | | |
|-----|---|--|-------|
| 1. |  | Program Flow | 5-4 |
| 2. |  | Move And Compare | 5-16 |
| 3. |  | Arithmetic And Logical Operations (+, -, ×, ÷) | 5-24 |
| 4. |  | Rotation And Shift | 5-34 |
| 5. |  | Data Operation | 5-42 |
| 6. |  | High Speed Processing | 5-52 |
| 7. |  | Handy Instructions | 5-66 |
| 8. |  | External FX I/O Devices | 5-80 |
| 9. |  | External FX Serial Devices | 5-94 |
| 10. |  | Floating Point 1 & 2 | 5-110 |

11.		Trigonometry (Floating Point 3)	5-118
12.		Data Operations 2	5-122
13.		Positioning Control	5-126
14.		Real Time Clock Control	5-136
15.		Gray Codes	5-146
16.		Additional Functions	5-146
		In-line Comparisons	5-150



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. D1,S3or for lists/abled devices D3+0,S+9etc.

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.

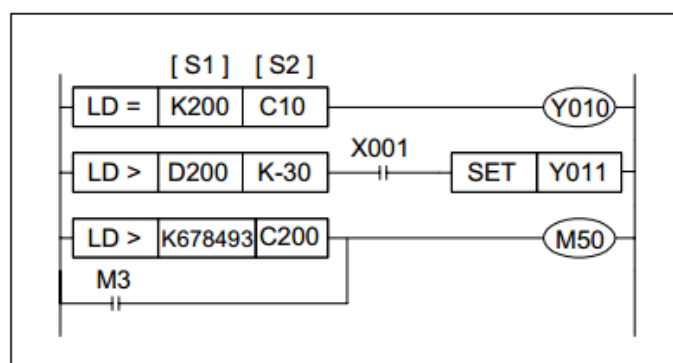
 - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

5.17.1 LD compare

(FNC 224 to 230)

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:

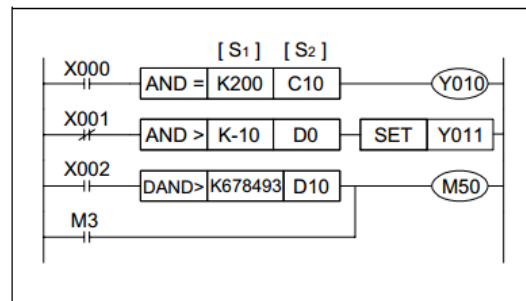
The LD comparison functions can be placed anywhere in a program that a standard LD instruction can be placed. I.e., it always starts a new block.

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)

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 S1 and S2 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:

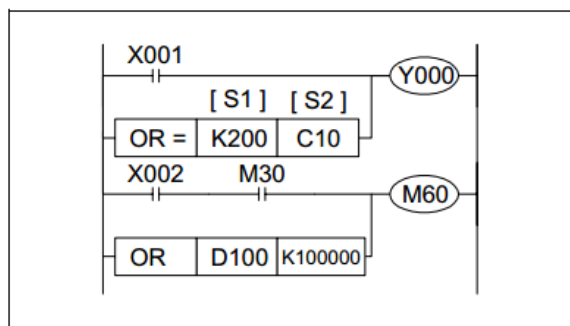
The AND comparison functions can be placed anywhere in a program that a standard AND instruction can be placed. i.e., it is a serial connection contact.

FNC No.	Mnemonic		Active when	Inactive when
	16 bit	32 bit		
232	AND =	DAND =	$S1 = S2$	$S1 \neq S2$
233	AND >	DAND >	$S1 > S2$	$S1 \leq S2$
234	AND <	DAND <	$S1 < S2$	$S1 \geq S2$
236	AND <>	DAND <>	$S1 \neq S2$	$S1 = S2$
237	AND ≤	DAND ≤	$S1 \leq S2$	$S1 > S2$
238	AND ≥	DAND ≥	$S1 \geq S2$	$S1 < S2$

5.17.3 OR compare (FNC 240 to 246)

Mnemonic	Function	Operands		Program steps
		S	D	
OR□ (OR compare) where □ is =, >, <, <>, ≤, ≥	Parallel comparison contact. Active when the comparison $S1 \square S2$ 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 S1 and S2 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.

FNC No.	Mnemonic		Active when	Inactive when
	16 bit	32 bit		
240	OR =	DOR =	$S1 = S2$	$S1 \neq S2$
241	OR >	DOR >	$S1 > S2$	$S1 \leq S2$
242	OR <	DOR <	$S1 < S2$	$S1 \geq S2$
244	OR <>	DOR <>	$S1 \neq S2$	$S1 = S2$
245	OR ≤	DOR ≤	$S1 \leq S2$	$S1 > S2$
246	OR ≥	DOR ≥	$S1 \geq S2$	$S1 < S2$

6. Diagnostic Devices

The following special devices are used by the PLC to highlight the current operational status and identify any faults or errors that may be occurring. There are some variations in the application of these devices to members of the PLC family, these are noted where appropriate.

The Internal diagnostic devices consist of both auxiliary (M) coils and data (D) registers. Often there is a correlation between both M and D diagnostic devices for example M8039 identifies that the PLC is in constant scan mode but D8039 contains the value or length of the set constant scan.



Devices unable to be set by user:

Any device of type M or D that is marked with a “()” cannot be set by a users program.

In the case of M devices this means the associated coil cannot be driven BUT all contacts can be read. For data devices (D) new values cannot be written to the register

by a user BUT the register contents can be used in a data comparison.

Default Resetting Devices:

- Certain devices reset to their default status when the PLC is turned from OFF to ON.

These are identified by the following symbol “(R)”

Symbol summary:

-  not able to be set by user
-  automatically reset to default at power ON.
-  Also reset to default when CPU is switched to RUN.
-  Also reset to default when CPU is switched to STOP.

6.1 Device Lists

Device	HCA1P	HCA2P/HCA2C
M8000	★	★
M8001	★	★
M8002	★	★
M8003	★	★
M8004	★	★
M8005	-	-
M8006	-	-
M8007	-	-

M8008	-	-
M8009	-	-
M8010	Reserved	
M8011	★	★
M8012	★	★
M8013	★	★
M8014	★	★
M8015	★	★
M8016	★	★
M8017	★	★
M8018	★	★
M8019	★	★
M8020	★	★
M8021	★	★
M8022	★	★
M8023	Reserved	
M8024	-	-
M8025	-	-
M8026	-	-
M8027	-	-
M8028	★*1	-

M8029	★	★
M8030	-	-
M8031	★	★
M8032	★	★
M8033	★	★
M8034	★	★
M8035	★	★
M8036	★	★
M8037	★	★
M8038	★	★
M8039	★	★
M8040	★	★
M8041	★	★
M8042	★	★
M8043	★	★
M8044	★	★
M8045	★	★
M8046	★	★
M8047	★	★
M8048	-	-
M8049	-	-

Device	HCA1P	HCA2P/HCA2C
D8000	★	★
D8001	★	★
D8002	★	★
D8003	★	★
D8004	★	★
D8005	-	-
D8006	-	-
D8007	-	-
D8008	-	-
D8009	-	-
D8010	★	★
D8011	★	★
D8012	★	★
D8013	★	★
D8014	★	★
D8015	★	★
D8016	★	★
D8017	★	★
D8018	★	★

D8019	★	★
D8020	★	★
D8021	Reserved	
D8022		
D8023		
D8024		
D8025		
D8026		
D8027		
D8028	★	★
D8029	★	★
D8030	★	★
D8031	★	★
D8032	Reserved	
D8033		
D8034		
D8035		
D8036		
D8037		
D8038		
D8039	★	★

D8040	★	★
D8041	★	★
D8042	★	★
D8043	★	★
D8044	★	★
D8045	★	★
D8046	★	★
D8047	★	★
D8048	Reserved	
D8049	-	-

Device	HCA1P	HCA2P/HCA2C
M8050	★	★
M8051	★	★
M8052	★	★
M8053	★	★
M8054	★	★
M8055	★	★
M8056	-	-
M8057	-	-

M8058	-	-
M8059	-	-
M8060	-	-
M8061	★	★
M8062	-	-
M8063	★	★
M8064	★	★
M8065	★	★
M8066	★	★
M8067	★	★
M8068	★	★
M8069	-	-
M8070	★	★
M8071	★	★
M8072	★	★
M8073	★	★
M8074	Reserved	
M8075	-	-
M8076	-	-
M8077	-	-
M8078	-	-

M8079	-	-
M8080	Reserved	
M8081		
M8082		
M8083		
M8084		
M8085		
M8086		
M8087		
M8088		
M8089		
M8090	Reserved	
M8091		
M8092		
M8093		
M8094		
M8095		
M8096		
M8097		
M8098		
M8099	-	-

Device	HCA1P	HCA2P/HCA2C
D8050	Reserved	
D8051		
D8052		
D8053		
D8054		
D8055		
D8056		
D8057		
D8058		
D8059		
D8060	-	-
D8061	★	★
D8062	-	-
D8063	★	★
D8064	★	★
D8065	★	★
D8066	★	★
D8067	★	★
D8068	★	★

D8069	★	★
D8070	★	★
D8071	Reserved	
D8072		
D8073		
D8074		
D8075	-	-
D8076	-	-
D8077	-	-
D8078	-	-
D8079	-	-
D8080	-	-
D8081	-	-
D8082	-	-
D8083	-	-
D8084	-	-
D8085	-	-
D8086	-	-
D8087	-	-
D8088	-	-
D8089	-	-

D8090	-	-
D8091	-	-
D8092	-	-
D8093	-	-
D8094	-	-
D8095	-	-
D8096	-	-
D8097	-	-
D8098	-	-
D8099	-	-

Device	HCA1P	HCA2P/HCA2C
M8100	Reserved	
M8101		
M8102		
M8103		
M8104		
M8105		
M8106		
M8107		
M8108		

M8109	-	-
M8110	Reserved	
M8111		
M8112		
M8113		
M8114		
M8115		
M8116		
M8117		
M8118		
M8119		
M8120	Reserved	
M8121	★	★
M8122	★	★
M8123	★	★
M8124	★	★
M8125	Reserved	
M8126	★	★
M8127	★	★
M8128	★	★
M8129	★	★

M8130	-	-
M8131	-	-
M8132	-	-
M8133	-	-
M8134	Reserved	
M8135		
M8136		
M8137		
M8138		
M8139		
M8140	★	★
M8141	Reserved	
M8142		
M8143		
M8144		
M8145	★	★
M8146	★	★
M8147	★	★
M8148	★	★
M8149	Reserved	
Device	HCA1P	HCA2P/HCA2C

D8100	Reserved	
D8101		
D8102	★	★
D8103	Reserved	
D8104		
D8105		
D8106		
D8107		
D8108		
D8109	-	-
D8110	Reserved	
D8111		
D8112		
D8113		
D8114		
D8115		
D8116		
D8117		
D8118		
D8119		
D8120	★	★

D8121	★	★
D8122	★	★
D8123	★	★
D8124	★	★
D8125	★	★
D8126	Reserved	
D8127	★	★
D8128	★	★
D8129	★	★
D8130	-	-
D8131	-	-
D8132	-	-
D8133	-	-
D8134	-	-
D8135	-	-
D8136	★	★
D8137	★	★
D8138	Reserved	
D8139		
D8140	★	★
D8141	★	★

D8142	★	★
D8143	★	★
D8144	Reserved	
D8145	★	★
D8146	★	★
D8147	★	★
D8148	★	★
D8149	Reserved	

Device	HCA1P	HCA2P/HCA2C
M8150	Reserved	
M8151		
M8152		
M8153		
M8154		
M8155		
M8156		
M8157		
M8158		
M8159		
M8160	-	-

M8161	★	★
M8162	★	★
M8163	Reserved	
M8164	-	-
M8165	Reserved	
M8166		
M8167	-	-
M8168	-	-
M8169	Reserved	
M8170	★	★
M8171	★	★
M8172	★	★
M8173	★	★
M8174	★	★
M8175	★	★
M8176	Reserved	
M8177		
M8178		
M8179		
M8180		
M8181		

M8182		
M8183	★M504	★
M8184	★M505	★
M8185	★M506	★
M8186	★M507	★
M8187	★M508	★
M8188	★M509	★
M8189	★M510	★
M8190	★M511	★
M8191	★M503	★
M8192	Reserved	
M8193		
M8194		
M8195		
M8196		
M8197		
M8198		
M8199		

Device	HCA1P	HCA2P/HCA2C
D8150	Reserved	

D8151		
D8152		
D8153		
D8154		
D8155		
D8156		
D8157		
D8158	★	★
D8159	★	★
D8160	Reserved	
D8161		
D8162		
D8163		
D8164	-	-
D8165	Reserved	
D8166		
D8167		
D8168		
D8169		
D8170		
D8171		

D8172		
D8173	★	★
D8174	★	★
D8175	★	★
D8176	★	★
D8177	★	★
D8178	★	★
D8179	★	★
D8180	★	★
D8181	Reserved	
D8182	★	★
D8183	★	★
D8184	★	★
D8185	★	★
D8186	★	★
D8187	★	★
D8188	★	★
D8189	★	★
D8190	★	★
D8191	★	★
D8192	★	★

D8193	★	★
D8194	★	★
D8195	★	★
D8196	Reserved	
D8197		
D8198		
D8199		

Note;

When using an N:N network configuration with the HCA1P, M503 to M511 are used in place of the regular M devices as shown above. D208 to D218 are used in place of the regular D devices shown below.

Device	HCA1P	HCA2P/HCA2C
M8200	-	★
M8201	-	★
M8202	-	★
M8203	-	★
M8204	-	★
M8205	-	★
M8206	-	★

M8207	-	★
M8208	-	★
M8209	-	★
M8210	-	★
M8211	-	★
M8212	-	★
M8213	-	★
M8214	-	★
M8215	-	★
M8216	-	★
M8217	-	★
M8218	-	★
M8219	-	★
M8220	-	★
M8221	-	★
M8222	-	★
M8223	-	★
M8224	-	★
M8225	-	★
M8226	-	★
M8227	-	★

M8228	-	★
M8229	-	★
M8230	-	★
M8231	-	★
M8232	-	★
M8233	-	★
M8234	-	★
M8235	★	★
M8236	★	★
M8237	★	★
M8238	★	★
M8239	★	★
M8240	★	★
M8241	★	★
M8242	★	★
M8243	★	★
M8244	★	★
M8245	★	★
M8246	★	★
M8247	★	★
M8248	★	★

M8249	★	★
M8250	★	★
M8251	★	★
M8252	★	★
M8253	★	★
M8254	★	★
M8255	★	★

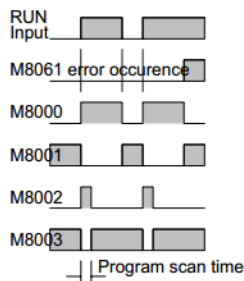
Device	HCA1P	HCA2P/HCA2C
D8200	Reserved	
D8201	★D201	★
D8202	★D202	★
D8203	★D203	★
D8204	★D204	★
D8205	★D205	★
D8206	★D206	★
D8207	★D207	★
D8208	★D208	★
D8209	★D209	★
D8210	★D210	★
D8211	★D211	★

D8212	★D212	★
D8213	★D213	★
D8214	★D214	★
D8215	★D215	★
D8216	★D216	★
D8217	★D217	★
D8218	★D218	★
D8219	Reserved	
D8220	Reserved	
D8221		
D8222		
D8223		
D8224		
D8225		
D8226		
D8227		
D8228		
D8229		
D8230	Reserved	
D8231		
D8232		

D8233	
D8234	
D8235	
D8236	
D8237	
D8238	
D8239	
D8240	Reserved
D8241	
D8242	
D8243	
D8244	
D8245	
D8246	
D8247	
D8248	
D8249	
D8250	Reserved
D8251	
D8252	
D8253	

D8254	
D8255	

6.2 PLC Status (M8000 to M8009 and D8000 to D8009)

Diagnostic Device	Operation
M8000 (X) RUN monitor NO contact	
M8001 (X) RUN monitor NC contact	
M8002 (X) Initial pulse NO contact	
M8003 (X) Initial pulse NC contact	
M8004 (X) Error	ON when one or more error

occurrence	flags from the range M8060 to M8067 are ON
M8005 (✗) Battery voltage Low (Not HCA1P/ HCA2P/HCA2C)	On when the battery voltage is below the value set in D8006
M8006 (✗) Battery error latch (Not HCA1P/ HCA2P/HCA2C)	Latches the battery Low error
M8007 (✗) Momentary power failure (Not HCA1P/ HCA2P/HCA2C)	See note 2
M8008 (✗) Power failure (Not HCA1P/	Power loss has occurred See note 2

HCA2P/HCA2C)	
M8009 (X)	Power failure of 24V
24V DC Down	DC
(Not HCA1P/	service supply
HCA2P/HCA2C)	

Diagnostic Device	Operation
D8000 (X)	HCA1P/ HCA2P/
Watchdog	HCA2C
timer	200ms
	See note 1
D8001 (X)	HCA1P:22
PLC type and	HCA2P/HCA2C:26
version	E.g. 26100 =
	HCA2P/HCA2C, V1.00
D8002 (X)	0002: 2K steps
Memory	(HCA1P only)
capacity	0008: 8K or 16k steps
(see also	(HCA2P/ HCA2C)

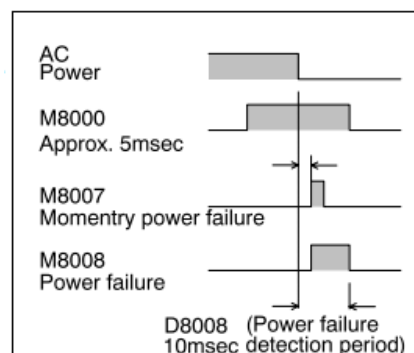
D8102)	
D8003 (✖) Memory type	00H = Option RAM, 01H = Option EPROM, 02H = Option EEPROM, 0AH = Option EEPROM(protected) 10H = Built-in MPU memory
D8004 (✖) Error number M☆☆☆☆	The contents of this register☆☆☆☆ identifies which error flag is active, i.e. if ☆☆☆☆= 8060 identifies M8060
D8005 (✖) Battery voltage (Not HCA1P/ HCA2P/HCA2C)	E.g. 36 = 3.6 volts
D8006 (✖)	The level at which a

Low battery voltage (Not HCA1P/ HCA2P/HCA2C)	low battery voltage is detected
D8007 (X) Power failure count(Not HCA1P/ HCA2P/HCA2C)	The number of times a momentary power failure has occurred since power ON
D8008 Power failure detection. (Not HCA1P/ HCA2P/HCA2C)	The time period before shut down when a power failure occurs (default 10ms) See note 2
D8009 (X) 24V DC failed device((Not HCA1P/ HCA2P/HCA2C)	Lowest device affected by 24V DC power failure



Note 1:

- The contents of this register can be changed by the user. Settings in 1 msec steps are possible. The value should be set greater than the maximum scan time (D8012) to ensure constant scan operation.



Note 2:

- When the power supply used is 200V AC, the power down detection period is determined by the value of D8008. This can be altered by the user within the allowable range of 10 to 100msec.

6.3 Clock Devices (M8010 to M8019 and D8010 to D8019)

Diagnostic Device	Operation
M8010	Reserved
M8011 (x) 10 msec clock pulse	Oscillates in 10 msec cycles
M8012 (x) 100 msec clock pulse	Oscillates in 100 msec cycles
M8013 (x) 1 sec clock pulse	Oscillates in 1 sec cycles
M8014 (x) 1 min clock pulse	Oscillates in 1 min cycles

Diagnostic Device	Operation
D8010 (x) Present scan time	Current operation cycle / scan time in units of 0.1 msec (waiting time for constant scan mode is included)
D8011 (x) Minimum scan time	Minimum cycle/ scan time in units of 0.1 msec (waiting time for constant scan mode is included)
D8012 (x) Maximum scan time	Maximum cycle/ scan time in units of 0.1 msec (waiting time for constant scan mode is included)

The following devices apply to HCA2C, HCA2P and HCA1P PLC's as standard when a real time clock option board installed.

M8015 Time setting	When ON - clock stops, ON ⇨ OFF restarts clock	D8013 Seconds	Seconds data for use with an RTC (0 - 59)
M8016 Register data	When ON D8013 to 19 are frozen for display but clock continues	D8014 Minute data	Minute data for use with an RTC (0-59)
M8017 Min. rounding	When pulsed ON set RTC to nearest minute	D8015 Hour data	Hour data for use with an RTC (0-23)
M8018 (X) RTC available	When ON Real Time Clock is installed	D8016 Day data	Day data for use with an RTC (1-31)
M8019 Setting error	Clock data has been set out of range	D8017 Month data	Month data for use with an RTC (1-12)
		D8018 Year data	Year data for use with an RTC (00-99 or 1980-2079, can be selected)
		D8019 Weekday data	Weekday data for use with an RTC (0-6)

6.4 Operation Flags (M8020 to M8029 and D8020 to D8029)

Diagnostic Device	Operation
M8020 (X) Zero	Set when the result of an ADD (FNC 20) or SUB (FNC 21) is “0”
M8021 (X) Borrow	Set when the result of a SUB (FNC 21) is less than the min. negative number
M8022 (F) Carry	Set when ‘carry’ occurs during an ADD (FNC 20) or when an overflow occurs as a

	result of a data shift operation
M8024 (Not HCA1P/ HCA2P/HCA2C)	BMOV (FNC 15) reverse mode. See note 3
M8025 (Not HCA1P/ HCA2P/HCA2C)	When ON HSC(FNC53-55) instructions are processed even when the external HSC reset input is activated
M8026 (Not HCA1P/ HCA2P/HCA2C)	RAMP (FNC 67) hold mode
M8027 (Not HCA1P/ HCA2P/HCA2C)	PR (FNC 77) 16 element data string
M8028	HCA1P: Change timers T32~T62to10mstype
M8029 (✕)	Set on the completion

Instruction execution complete	of operations such as DSW(FNC 72), RAMP (FNC 67) etc.
--	---

Diagnostic Device	Operation
D8020( See note 4	Input filter setting for devices; X000 to X017 default value = 10 msec, zero value = 50μsec (X000, X001: 20μsec) X000 to X007(HCA1P/ HCA2P/HCA2C) default value = 10msec zero value = 50μsec (X000, X001: 10μsec)
D8021( (Not HCA2P/HCA2C,) See note 4	Input filter setting for devices; X010 to X017 (HCA1P) default value = 10 msec,

	zero value = 50μsec
D8022 -D8027	Reserved
D8028 (X)	Current value of the Z0 index register See note 5
D8029 (X)	Current value of the V0 index register See note 5



Note 3

- If M8024 is used with a BMOV (FNC 15) instruction, it will operate as follows; M8024 OFF - Normal operation (Forwarding direction is [S] to [D]) M8024 ON - Reverse operation (Forwarding direction becomes [D] to [S]) This device is not supported in HCA1P and HCA2P/HCA2C.

Note 4

- The settings for input filters only apply to the main processing units which use 24V DC inputs. AC input filters are not adjustable.

6.5 PLC Operation Mode (M8030 to M8039 and D8030 to D8039)

Diagnostic Device	Operation	Diagnostic Device	Operation
M8030 (✗) Battery LED OFF	Battery voltage is low but BATT.V LED not lit	D8030 (X) 	Value read from first setting "pot" in msec, (0 to 255)
M8031 (✗) Non-latch memory all clear	Current device settings are reset at next END, i.e. contacts, coils and current data values for Y, M, S, T, C and D devices respectively.	D8031 (X) 	Value read from second setting "pot" in msec, (0 to 255)
M8032 (✗) Latch memory all clear	Special devices and file registers which have default settings are refreshed with those defaults	D8032 -D8038	Reserved
M8033 (✗) Memory hold in 'stop' mode	The device statuses and settings are retained when the PLC changes from RUN to STOP and back into RUN		
M8034 (✗) All outputs disable	All of the physical switch gear for activating outputs is disabled. However, the program still operates normally.		
M8035 (✗S) Forced operation mode	By using forced operation mode, i.e.M8035 is turned ON, it is possible to perform remote RUN/STOP or pulsed RUN/ STOP operation. Please see Chapter 10 for example operation		
M8036 (✗S) Forced RUN signal			
M8037 (✗S) Forced STOP signal			
M8038 N to N networking	For the setting of devices when using an N to N network	D8039 (✗) Constant scan duration	This register can be written to by the user to define the duration of the constant scan. Resolutions of 1msec are possible. This register has a default setting 0 msec which will be initiated during power ON.
M8039 (✗) Constant scan mode	When ON the PLC executes the user program within a constant scan duration. The difference between the actual end of the program operation and the set constant scan duration causes the PLC to 'pause'.		

6.6 Step Ladder (STL) Flags (M8040 to M8049 and D8040 to D8049)

Diagnostic Device	Operation	Diagnostic Device	Operation
M8040 (↯) STL transfer disable	When ON STL state transfer is disabled	D8040 (X) Lowest active STL step	Up to 8 active STL states, from the range S0 to S899, are stored in D8040 to D8047 in ascending numerical order. (Updated at END)
M8041 (↯S) Transfer start	When ON STL transfer from initial state is enabled during automatic operation (ref. IST FNC 60)	D8041 (X) 2nd active STL state	
M8042 (↯) Start pulse	A pulse output is given in response to a start input (ref. IST FNC 60)	D8042 (X) 3rd active STL state	
M8043 (↯S) Zero return complete	ON during the last state of ZERO RETURN mode (ref. IST FNC 60)	D8043 (X) 4th active STL state	
M8044 (↯S) Zero point condition	ON when the machine zero is detected (ref. IST FNC 60)	D8044 (X) 5th active STL state	
M8045 (↯) All output reset disable	Disables the 'all output reset' function when the operation mode is changed (ref. IST FNC 60)	D8045 (X) 6th active STL state	
M8046 (X) STL state ON	ON when STL monitoring has been enabled (M8047) and there is an active STL state	D8046 (X) 7th active STL state	
M8047 (↯) Enable STL monitoring	When ON D8040 to D8047 are enabled for active STL step monitoring	D8047 (X) 8th active STL state	
M8048 (X) Annunciator ON (Not FX1S, FX1N)	ON when Annunciator monitoring has been enabled (M8049) and there is an active Annunciator flag	D8048	Reserved
M8049 (↯) Enable Annunciator monitoring (Not FX1S, FX1N)	When ON D8049 is enabled for active Annunciator state monitoring	D8049 (X) Lowest active Annunciator (Not FX1S, FX1N)	Stores the lowest currently active Annunciator from the range S900 to S999 (Updated at END)



General note:

- M8046 to M8049 STL states are updated when the END instruction is executed.

6.7 Interrupt Control Flags (M8050 to M8059 and D8050 to D8059)

Diagnostic Device	Operation
M8050 (※) I00□ disable	When the EI (FNC 04) instruction is driven in the user program, all interrupts are enabled unless the special M devices noted here are driven ON. In that case for each special M coil that is ON, the associated interrupt is disabled, i.e. will not operate. Note □□ denotes all types of that interrupt
M8051 (※) I10□ disable	
M8052 (※) I20□ disable	
M8053 (※) I30□ disable	
M8054 (※) I40□ disable	
M8055 (※) I50□ disable	
M8056 (※) I6□□ disable	
M8057 (※) I7□□ disable	
M8058 (※) I8□□ disable	I010 ~ I060 is disabled for high speed counter interrupt (FNC53) When this flag is ON, the associated interrupt is disabled and therefore will not operate.
M8059(※) I010 to I060 disabled as a single group	

Diagnostic Device	Operation
D8050 -D8059	Reserved

6.8 Error Detection Devices (M8060 to M8069 and D8060 to D6069)

Diagnostic Device	Operation					Diagnostic Device	Operation
	ON- OFF	OFF- ON	Other	PROG LED	PLC STATUS		
M8060 (X) I/O configuration error	✓	✓	While the PLC is in RUN	OFF	RUN	D8060 (X)	The first I/O number of the unit or block causing the error - See note 6
M8061 (X) PLC hardware error	✓	-		ON	STOP	D8061 (X)	Error code for hardware error - See appropriate error code table
M8062 (X) PC/HPP comms error on programming port	-	-	When a signal from the programming port is received	OFF	RUN	D8062 (X)	Error code for PC/HPP Communications error - See appropriate error code table
M8063(X)(↯R) Parallel link/ RS232-C and RS485 (422) comms error on optional port	-	-	When a signal from the optional port is received			D8063(X)(↯R)	Error code for parallel link error - See communication users manual
M8064 (X) Parameter error	✓	✓	When the program is changed (PLC in STOP) and when a program is transferred (PLC in STOP)	Flash	STOP	D8064 (X)	Error code identifying parameter error - See appropriate error code table
M8065 (X) Syntax error						D8065 (X)	Error code identifying syntax error - See appropriate error code table
M8066 (X) Program error						D8066 (X)	Error code identifying program construction error - See appropriate error code table
M8067(X)(↯R) Operation error	-	-	While in PLC is in RUN	OFF	RUN	D8067(X)(↯R)	Error code identifying operation error. See appropriate error code table
M8068 (↯) Operation error latch						D8068 (↯)	Operation error step number latched
M8069 (↯) I/O bus error			See note 7	-	-	D8069(X)(↯R)	Step numbers for found errors corresponding to flags M8065 to M8067

② • Please see the following page for the notes referenced in this table.

① Note 6:

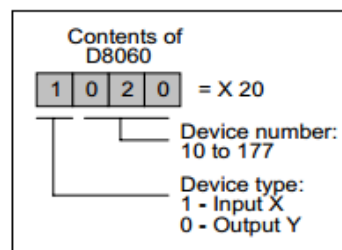
• If the unit or block corresponding to a programmed I / O number is not actually loaded, M8060 is set to ON and the first device number of the erroneous block is written to D8060.

Note 7:

•An I/O bus check is executed when M8069 is turned ON.

If an I/O bus error occurs, error code 6103 is written to

D8069 and M8061 is turned ON.



If an Extension unit 24V failure occurs, error code 6104 is written to D8061 and M8061

is turned ON. M8009 will then be turned ON and the I/O address of the lowest

numbered device affected by the 24V DC power failure is written to D8009

General note:

•HPP refers to Handy programming panel.

6.9 Link and Special Operation Devices (M8070 to M8099 and D8070 to D8099)

Diagnostic Device	Operation	Diagnostic Device	Operation
M8070 (↔R)	Driven when the PLC is a master station in a parallel link application	D8070 (X)	Parallel link watchdog time - 500 msec
M8071 (↔R)	Driven when the PLC is a slave station in a parallel link application	D8071 - D8073	Reserved
M8072 (X)	ON while the PLC is operating in a parallel link		
M8073 (X)	ON when M8070/ M8071 are incorrectly set during parallel link operations		
M8074	Reserved		
M8075	When executing Sampling trace in HCP-Works , these devices are used by the PLC internal system	D8074	When executing Sampling trace in HCP-Works , these devices are used by the PLC internal system
M8076		D8075	
M8077		D8076	
M8078	ON during sampling trace	D8077	
M8079	ON when sampling trace complete	D8078	
M8080 -M8098	Reserved	D8079	
		D8080 to D8095	
		D8096 to D8098	
		D8099	
M8099 (↔)	High speed free timer operation When ON, continue counting free ring timer (D8099)		Free ring timer, range: 0-32,767 in units of 0.1 msec (for use in measuring high speed pulse input durations) See section 10.9.2

6.10 Miscellaneous Devices

(M8100 to M8119 and D8100 to D8119)

Diagnostic Device	Operation
M8109 (X) Not HCA1/HCA2 (C)	Output refresh error

Diagnostic Device	Operation
D8102 (X) Memory Capacity	0002: 2K steps (HCA1) 0004: 4K steps (HCA5) 0008: 8K steps (HCA2(C)) 0016: 16K steps (HCA5)
D8109 (X)	Output refresh error, lowest device number; 0, 10, 20, etc.

6.11 Communication Adapter

Devices, i.e. 232ADP, 485ADP (M8120 to M8129 and D8120 to D8129)

Diagnostic Device	Operation
M8120	Reserved
M8121(X)(↔R)	Data transmission delayed (RS instruction)
M8122 (↔R)	Data transmission flag (RS instruction)
M8123 (↔R)	Finished receiving data (RS instruction)
M8124(X)	Carrier detection flag (RS instruction)
M8125	Reserved
M8126	Global flag (Computer link)
M8127 (↔)	On Demand handshake flag (Computer link)
M8128 (↔)	On Demand error flag (Computer link)
M8129 (↔)	On Demand Byte/Word change over (Computer link), Time out evaluation flag (RS instruction)

Diagnostic Device	Operation
D8120	Communications format (RS instruction, Computer link)
D8121	Station number setting (Computer link)
D8122(X)(↔R)	Amount of remaining data to be transmitted (RS instruction)
D8123(X)(↔R)	Amount of data already received (RS instruction)
D8124 (↔)	Data header, default STX (02H) (RS instruction)
D8125 (↔)	Data terminator, default ETX (03H) (RS instruction)
D8126	Reserved
D8127 (↔)	On Demand head device register (Computer link)
D8128 (↔)	On Demand data length register (Computer link)
D8129	Data network 'time-out' timer value (RS instruction, Computer link)

6.12 High Speed Zone Compare Table Comparison Flags

(M8130 to M8148 and D8130 to D8148)

Diagnostic Device	Operation	Diagnostic Device	Operation
M8130 See note 8	Selects comparison tables to be used with the HSZ instruction	D8130 (X)(∞)	Contains the number of the current record being processed in the HSZ comparison table
M8131 (X)(∞) See note 8	ON when the HSZ comparison table has been completed.	D8131 (X)(∞)	Contains the number of the current record being processed in the HSZ comparison table when the PLSY operation has been enabled
M8132 See note 8	Selects the use of the PLSY instruction with the HSZ comparison tables	D8132 D8133 (X)(∞)	Contains the source (output pulse frequency) data for the PLSY instruction when used with the HSZ comparison table
M8133 (X)(∞) See note 8	ON when the HSZ comparison table (when used with the PLSY instruction) has been completed.	D8134 D8135 (X)(∞)	Contains a copy of the value for the current comparison when the HSZ comparison table and combined PLSY output are used. This data is only available in 32 bit or double word format.
M8134- M8139	Reserved	D8136 D8137 (X)(∞)	Contains the total number of pulses that have been output using the PLSY (or PLSR) instruction on Y000 and Y001. This data is only available in 32 bit or double word format
		D8138 - D8139	Reserved

Note 8

- See section 5.6.6 for full explanation and use.

Diagnostic Device	Operation	Diagnostic Device	Operation
M8140 (X) (✗)	When ON, clears pulse output in FNC156(ZRN) instruction	D8140 D8141 (X) (✗)	Contains the total number of pulses that have been output to Y0 using the PLSY or PLSR instructions. This data is only available in 32 bit or double word format.
M8141 to M8144	Reserved	D8142 D8143 (X) (✗)	Contains the total number of pulses that have been output to Y1 using the PLSY or PLSR instructions. This data is only available in 32 bit or double word format.
M8145 (✗)	Y000 Pulse output stop command	D8145 (✗)	FNC156(ZRN), FNC158(DRV), FNC159(DRVA) Bias value setting (default:0)
M8146 (✗)	Y001 Pulse output stop command	D8146 (✗)	FNC156(ZRN), FNC158(DRV), FNC159(DRVA)
M8147 (X)	Y000 Pulse output monitor (Busy/Ready)	D8147 (✗)	Max. speed setting (default:100,000)
M8148 (X)	Y001 Pulse output monitor (Busy/Ready)	D8148 (✗)	FNC156(ZRN), FNC158(DRV), FNC159(DRVA) Acceleration/Deceleration time setting (default:100)

6.13 Miscellaneous Devices (M8160 to M8199)

Diagnostic Device	Operation
M8160 (↯)	Selection of XCH operation to swap bytes in a single data word
M8161 (↯)	Selection of 8 bit operations for applied instructions ASC, RS, ASCI, HEX, CCD
M8162 (↯)	High speed mode for Parallel link, 2 data words Read/write only
M8164 (↯)	When ON, a value in D8164 is used as the number of FROM/TO exchange points. (FX2N/CPU Version 2.00 and above)
M8167 (↯)	Selection of hexadecimal input mode for the HKY instruction
M8168 (↯)	Selection of BCD mode for use with the SMOV instruction
M8169	Reserved
M8170 (↯R) X0 pulse catch	When the leading edge of a pulse is received at an input from the range X0 to X5 the associated M device detailed here is set ON. By resetting the same device within the user program the next pulse occurrence will again set the M coil ON. Hence, fast input pulses are 'caught' and stored. This operation requires the EI (FNC04) instruction to be active. For details see page 6-12
M8171 (↯R) X1 pulse catch	
M8172 (↯R) X2 pulse catch	
M8173 (↯R) X3 pulse catch	
M8174 (↯R) X4 pulse catch	
M8175 (↯R) X5 pulse catch	

Diagnostic Device	Operation
M8176 -M8199	Reserved

6.14 Miscellaneous devices (D8158 to D8164) and Index Registers (D8182 to D8199)

Diagnostic Device	Operation
D8158 (✕)	Control device for HCA2-5DM*1 Default: k-1
D8159 (✕)	Control device for HCA2-5DM*1 Default: k-1
D8164 (✕)	Number of FROM/TO exchange points (HCA5 CPU Version 2.00 and above)
D8181 (X)	Reserved
D8182 (X)	Value of Z1 index register
D8183 (X)	Value of V1 index register
D8184 (X)	Value of Z2 index register
D8185 (X)	Value of V2 index register
D8186 (X)	Value of Z3 index register

Diagnostic Device	Operation
D8187 (X)	Value of V3 index register
D8188 (X)	Value of Z4 index register
D8189 (X)	Value of V4 index register
D8190 (X)	Value of Z5 index register
D8191 (X)	Value of V5 index register
D8192 (X)	Value of Z6 index register
D8193 (X)	Value of V6 index register
D8194 (X)	Value of Z7 index register
D8195 (X)	Value of V7 index register

*1 See Chapter 10.19.2 for more information

6.15 N:N Network Related Flags and Data Registers

Diagnostic Device	Operation
D8173 (X)	Station number
D8174 (X)	Total number of slave stations
D8175 (X)	Refresh range
D8176 See note10	Station number setting Default value k0
D8177 See note10	Total number of slave stations setting

	Default value k7
D8178	Refresh range setting
See note10	Default value k0
D8179	Retry count setting
See note10	Default value k3
D8180	Comms time-out setting
See note10	Default value k5
8201 (✖) (For HCA1Puse D201)	Current network scan time
D8202 (✖) (For HCA1Puse D202)	Maximum network scan time
D8203 (✖) (For HCA1Puse D203)	Number of communication error at master station
D8204 to D8210 (✖) (For HCA1Puse D204 to D210)	Number of communication error at respective slave station
D8211 (✖) (For HCA1Puse D2113)	Code of communication error at master station
D8212 to D8218 (✖)	Code of communication

(For HCA1Puse D212 to D218)	error at respective slave station	Diagnostic Device	Operation
		M8183 (X) (For HCA1Puse M504	ON when communication error in master station
		M8184 (X) (For HCA1Puse M505	ON when communication error in 1 st slave station
		M8185 (X) (For HCA1Puse M506)	ON when communication error in 2 nd slave station
		M8186 (X) (For HCA1Puse M507	ON when communication error in 3 rd slave station
		M8187 (X) (For	ON when communication error

HCA1Puse M508	in 4 th slave station
M8188 (X) (For HCA1Puse M509)	ON when communication error in 5 th slave station
M8189 (X) (For HCA1Puse M510)	ON when communication error in 6 th slave station
M8190 (X) (For HCA1Puse M511)	ON when communication error in 7 th slave station
M8191 (X) (For HCA1Puse M503)	ON when communicating to another station



Note 9

- Devices M503-M511 and D201-D255 in the HCA1P cannot be applied to other functions in the user program. These devices are used exclusively for the N:N Network.

Note 10

- When these devices are not being used for an N:N Network their respective default values are all '0'. The relevant default values are assumed at each power ON.

6.16 Up/Down Counter Control (M8200 to M8234 and D8219 to D8234)

Diagnostic Device	Operation	Diagnostic Device	Operation
M8200 - M8234 (≠)	When M8☆☆ is operated, counter C☆☆ functions as a down counter. When M8☆☆ is not operated the associated counter operates as an up counter	D8219 -D8234	Reserved

6.17 High Speed Counter Control (M8235 to M8255 and D8235 to D8255)

Diagnostic Device	Operation	Diagnostic Device	Operation
M8235 -M8245 (≠)	When M8☆☆ is operated, the 1 phase high speed counter C☆☆ functions as a down counter. When M8☆☆ is not operated the associated counter operates as an up counter. The available counters depends upon the PLC type.	D8235 -D8255	Reserved
M8246 - M8255 (X)(≠)	When M8☆☆ is operated, the 2 phase high speed counter C☆☆ functions as a down counter. When M8☆☆ is not operated the associated counter operates as an up counter. The available counters depends upon the PLC type.		

6.18 Error Code Tables

Error Detection Device	Stored Error Number	Associated Meaning	Action
D8061 PLC Hardware error	0000	No error	Check the cable connection between the extension unit/block and the PLC
	6101	RAM error	
	6102	Operation circuit error	
	6103	I/O bus error (M8069 = ON)	
	6104	Extension unit 24V failure (M8069=ON)	Scan time has exceeded the WDT time value set in D8000. Check user program.
	6105	Watch Dog Timer error	

Error Detection Device	Stored Error Number	Associated Meaning	Action
D8062 PC/HPP communication error	0000	No error	Check the cable connection between the programming device and the PLC
	6201	Parity/ overrun/ framing error	
	6202	Communications character error	
	6203	Communication data sum check error	
	6204	Data format error	
	6205	Command error	

Error Detection Device	Stored Error Number	Associated Meaning	Note
D8063 Serial communication errors	0000	No error	Check communication settings, parameters and applicable devices. (Computer link, N:N network, Parallel link etc.) Refer to Communication Users Manual for wiring techniques
	6301	Parity/ overrun/ framing error	
	6302	Comms character error	
	6303	Comms data sum check error	
	6304	Comms data format error	
	6305	Command error Computer link - received command other than GW (global) when station number was FF	
	6306	Watchdog timer error	
	6312	Parallel link character error	
	6313	Parallel link data sum check error	
	6314	Parallel link data format error	

Error Detection Device	Stored Error Number	Associated Meaning	Action
D8064 Parameter error	0000	No error	STOP the PLC, check parameter, if incorrect change to a suitable value
	6401	Program sum check error	
	6402	Memory capacity setting error	
	6403	Latched device area setting error	
	6404	Comment area setting error	
	6405	File register area setting error	
	6406 - 6408	Reserved	
	6409	Other setting error	

Error Detection Device	Stored Error Number	Associated Meaning	Action
D8065 Syntax error	0000	No error	During programming, each instruction is checked as it is entered. If a syntax error is detected, re-enter the instruction correctly
	6501	Incorrect instruction/ device symbol/ device number combination	
	6502	No timer or counter coil before setting value	
	6503	1)No setting value following either a timer or a counter coil 2)Insufficient number of operands for an applied instruction	
	6504	1)The same label number is used more than once 2)The same interrupt input or high speed counter input is used more than once	
	6505	Device number is outside the allowable range	
	6506	Invalid applied instruction	
	6507	Invalid Pointer device [P] assignment for Jump or Call instruction	
	6508	Invalid Interrupt pointer device [I] assignment	
	6509	Other error	
	6510	MC nesting (N) number error	
	6511	The same interrupt input or high speed counter input is used more than once	

Error Detection Device	Stored Error Number	Associated Meaning	Action
D8066 Circuit error	0000	No error	A circuit error occurs if a combination of instructions is incorrect or badly specified. Select programming mode and correct the identified error.
	6601	LD and LDI is used continuously 9 or more times in succession	
	6602	1)No LD/ LDI instruction. The use of LD/LDI or ANB/ORB instruction is incorrect. 2)The following instructions are not connected to the active bus line: STL, RET, MCR, (P)ointer, (I)nterrupt, EI, DI,	

		SRET, IRET, FOR, NEXT, FEND and END 3)When MPP is missing	
	6603	MPS is used continuously more than 12 times	
	6604	The use of MPS, MRD, MPP instruction is incorrect	
	6605	1)The STL instruction is continuously used 9 times or more 2)MC, MCR instruction, (I)nterrupt pointer orSRETinstructionisusedwithinanSTL program area 3)RET has not been used in the program or is not connected to an STL instruction	
	6606	1)No (P)ointer, (I)nterrupt pointer 2)No SRET/ IRET 3)An (I)nterrupt pointer, SRET or IRET has been used within the main program	

		4)STL, RET, MC or MCR have been used within either a subroutine or an interrupt routine	
	6607	1)The use of FOR and NEXT is incorrect 2)The following instructions have been used within a FOR -NEXT loop: STL, RET, MC, MCR, IRET, SRET, FEND or END	
	6608	1)The use of MC/ MCR is incorrect 2)Missing MCR N0 3)SRET, IRET instruction or an (I)nterrupt pointer has been used within an MC/ MCR instruction area	
	6609	Other error	

Error Detection Device	Stored Error Number	Associated Meaning	Action
D8066 Circuit error	6610	LD, LDI is used continuously 9 or more times in succession	A circuit error occurs if a combination of instructions is incorrect or badly
	6611	Number of LD/LDI instructions is more than ANB/ORB instructions	

	6612	Number of LD/LDI instructions is less than ANB/ORB instructions	specified. Select programming mode and correct the identified error.
	6613	MPS is used continuously more than 12 times	
	6614	MPS instruction missing	
	6515	MPP instruction missing	
	6616	Unauthorized use of the MPS/ MRD/ MPP instructions; possible coil missing	
	6616	Unauthorized use of the MPS/ MRD/ MPP instructions; possible coil missing	
	6617	One of the following instructions is not connected to the active bus line: STL, RET, MCR, (P)ointer, (I)nterrupt pointer, EI, DI, SRET, IRET, FOR, NEXT, FEND and END	
	6618	STL, RET, MC or MCR programmed within either a subroutine or an interrupt routine	
	6619	Invalid instruction programmed within a FOR - NEXT loop: STL, RET, MC, MCR, (I)nterrupt pointer, IRET and SRET	

	6620	FOR - NEXT instruction nesting levels (5) exceeded	
	6621	The number of FOR and NEXT instructions does not match	
	6622	NEXT instruction not found	
	6623	MC instruction not found	
	6624	MCR instruction not found	
	6625	The STL instruction is continually used 9 times or more	
	6626	Invalid instruction programmed within an STL - RET program area: MC, MCR, (I)nterrupt pointer, IRET and SRET	
	6627	RET instruction not found	
	6628	(I)nterrupt pointer, SRET and IRET incorrectly programmed within main program	
	6629	(P)ointer or (I)nterrupt pointer label not found	
	6630	SRET or IRET not found	
	6631	SRET programmed in invalid location	
	6632	IRET programmed in invalid location	

Error Detection Device	Stored Error Number	Associated Meaning	Action
D8067 Operation error	0000	No error	These error occur during the execution of an operation. When an operation error occurs, STOP the PLC enter programming ode and correct the fault. Note: operation errors can occur even when the syntax or circuit design is correct, e.g. D500Z is a valid statement within an HCA2P/HCA2C PLC. But if Z had a value of 10000, the data
	6701	1)No jump destination (pointer) for CJ or CALL instructions 2)(P)ointer is designated in a block that comes after the END instruction 3)An independent label is designated in a FOR-NEXT loop or a subroutine	
	6702	6 or more CALL instruction nesting levels have been used	
	6703	3 or more interrupt nesting levels have been used	
	6704	6 or more FOR - NEXT instruction nesting levels have been used	
	6705	An incompatible device has been specified as an operand for an applied instruction	
	6706	A device has been specified outside of	

		the allowable range for an applied instruction operand	register D10500 would be attempted to be accessed. This will cause an operation error as there is no D10500 device available.
	6707	A file register has been accessed which is outside of the users specified range	
	6708	FROM/ TO instruction error	
	6709	Other error, i.e. missing IRE/SRET, unauthorized FOR - NEXT relationship	
D8067 PID Operation error	6730	Sampling time TS ($TS < 0$ or > 32767)	The identified parameter is specified outside of its allowable range Execution ceases PID instruction must be reset before execution will resume
	6732	Input filter value α ($\alpha < 0$ or ≥ 101)	
	6733	Proportional gain KP ($KP < 0$ or > 32767)	
	6734	Integral time constant TI ($TI < 0$ or > 32767)	
	6735	Derivative gain KD ($KD < 0$ or ≥ 101)	
	6736	Derivative time constant TD ($TD < 0$ or > 32767)	TS is set to program scan time - Execution will continue
	6740	Sampling time TS is less than the program scan time	
	6742	Current value Δ exceeds its limits	
	6743	Calculated error exceeds its limits	
	6744	Integral result exceeds its limits	For all errors except

	6745	Derivative gain over, or differential value exceeds allowable range	6745, this will either be a minimum of -32768 or a maximum of +32767. Execution will continue, but user should reset PID instruction.
	6746	Derivative result exceeds its limits	
	6747	Total PID result exceeds its limits	
	6750	SV - PVnf < 150, or system is unstable (SV - PV nf has wide, fast variations)	The error fluctuation is outside the normal operation limits for the PID instruction. Execution ceases. PID instruction must be reset.
	6751	Large Overshoot of the Set Value	
	6752	Large fluctuations during Autotuning Set Process	

7. Execution Times And Instructional Hierarchy

7.1 Basic Instructions

Mnemonic	Object Devices	Steps	Execution Time in μsec			
			HCA1P		HCA2P/HCA2C	
			ON	OFF	ON	OFF

LD	X,Y,M,S,T,C and special M	1	0.7			
LDI						
AND			0.65			
ANI						
OR						
ORI						
LDP	X,Y,M,S,T,C	1	11.7	-	11.7	-
LDF				-		-
ANDP				-		-
ANDF				-		-
ORP				-		-
ORF				-		-
ANB	Not applicable	1	0.55			
ORB						
MPS			0.5			
MRD			0.55			
MPP			0.5			
INV						
MC	Nest level, M,Y	3	8.6	8.0	8.6	8.0
MCR	Nest leve	2	4.1	-	4.1	-

NOP	Not	1	0.45			
END	applicable		450	-	450	-
STL	S	1	15.8+	-	15.8+	-
	(see note 1)		8.2n		8.2n	
RET	Not		4.8	-	4.8	-
	applicable					

Mnemonic	Object Devices	Steps	Execution Time in μ sec			
			HCA1P		HCA2P/HCA2C	
			ON	OFF	ON	OFF
OUT	Y, M	1	0.7			
	S	2	4.4			
	Special M	2	2.8			
	T-K	3	11.2	10.2	11.2	10.2
	T-D	3	12.2	11.2	12.2	11.2
	C-K (16 bit)	3	8.1	6.9	8.1	6.9
	C-D (16 bit)	3	9.5	8.0	9.5	8.0
	C-K (32 bit)	5	8.1	6.8	8.1	6.8
	C-D (32 bit)	5	9.5	8.0	9.5	8.0
SET	Y, M	1	0.85			
	S	2	4.2	2.4	4.2	2.4

	S when used in an STL step (see note 1)		18.6+	2.4	18.6+	2.4
	Special M		6.8n		6.8n	
			2.8			
RST	Y, M	1	0.85			
	S	2	3.8	2.4	3.8	2.4
	Special M	2	2.8			
	T, C	2	8.7	7.3	8.7	7.3
	D, V, Z and special D	3	3.8	1.1	3.8	1.1
PLS	Y, M	2	10.8			
PLF	Y, M	2				
P	OTO6	1	0.45			
I	I□□□	1				



Note 1:

- “n” in the formulae to calculate the ON/OFF execution time, refers to the number of STL instructions at the current parallel/merge branch. Thus the value of “n” will fall in the range 1 to 8.

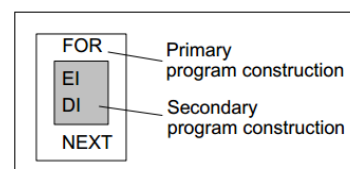
7.2 Hierarchical Relationships Of Basic Program Instructions

The following table identifies an 'inclusive relationship'.

This means the secondary program construction is

included within the complete operating boundaries of

the primary program construction, e.g.:



Primary Program Construction	Secondary program construction							
	MC-MCR	CJ - P	EI - DI	FOR - NEXT	STL - RET	P - SRET	I - IRET	FEND - END
MC - MCR	✓ - 8 nest levels	✓	✓	✓	✓	x - (6608)	x - (6608)	x - (6608)
CJ - P	✓	✓	✓	✓	✓	●	●	x - (6701)
EI - DI	✓	✓	✓	✓	✓	✓	✓	✓①
FOR - NEXT	x - (6607)	✓	✓	✓ - 5 nest levels	x - (6607)	x - (6607)	x - (6607)	x - (6607)
STL - RET	x - (6605)	●	✓	✓ - (within 1 STL step)	✓	x - (6605)	x - (6605)	x - (6605)
P - SRET	x - (6606)	✓	✓	✓	x - (6606)	x - (6606)	x - (6606)	x - (6709)
I - IRET	x - (6606)	✓	✓	✓	x - (6606)	x - (6606)	x - (6606)	x - (6606)
FEND - END	✓	✓	✓	✓	●	✓	✓	✓②
0 - FEND	✓	✓	✓	✓	✓	x - (6606)	x - (6606)	✓②
0 - END (no FEND)	✓	✓	✓	✓	✓	x - (6606)	x - (6606)	✓②

  Instruction combination is acceptable - for restrictions see appropriate note.

 Instruction combination is not allowed - bracketed number is the error code.

 Instruction combination is not recommended for use even though there is no operational error.

The combination of instructions with an 'inclusive relationship' is allowable. However please be aware of the following exceptions:

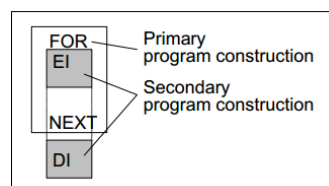
1) MC-MCR and STL-RET constructions cannot be used within FOR-NEXT loops, P-SRET or I-IRET subroutines.

2) Program flow may not be discontinued by using any of the following methods while

inside MC-MCR, FOR-NEXT, P-SRET, I-IRET program

constructions, i.e. using interrupts (I), IRET, SRET,

FEND or the END instruction is not allowed.



The following table identifies an 'overlapping relationship'. This means the secondary

program

construction starts within the complete operating boundaries of the primary program

construction but finishes outside of the primary construction, e.g.:

Primary Program Construction	Secondary program construction							
	MC-MCR	CJ - P	EI - DI	FOR - NEXT	STL - RET	P - SRET	I - IRET	FEND - END
MC - MCR	●	●	✓	X - (6607)	X - (6605)	X - (6606)	X - (6606)	X - (6608)
CJ - P	●	●	✓	●	●	●	●	✓
EI - DI	✓	✓	✓	✓	✓	✓	✓	✓
FOR - NEXT	X - (6607)	●	✓	✓ ^①	X - (6601)	X - (6607)	X - (6607)	X - (6607)
STL - RET	X - (6605)	●	✓	X - (6607)	✓	X - (6606)	X - (6606)	X - (6605)
P - SRET	X - (6608)	●	✓	X - (6607)	X - (6605)	X - (6606)	X - (6606)	X - (6709)
I - IRET	X - (6606)	●	✓	X - (6607)	X - (6606)	X - (6606)	X - (6606)	X - (6606)
FEND - END	X - (6608)	X - (6601)	✓ ^①	X - (6607)	X - (6605)	X - (6709)	X - (6709)	✓ ^②
0 - FEND	X - (6608)	✓	✓	X - (6607)	X - (6605)	X - (6709)	X - (6606)	✓ ^②
0 - END (no FEND)	X - (6608)	X - (6601)	✓ ^①	X - (6607)	X - (6605)	X - (6709)	X - (6606)	✓ ^②

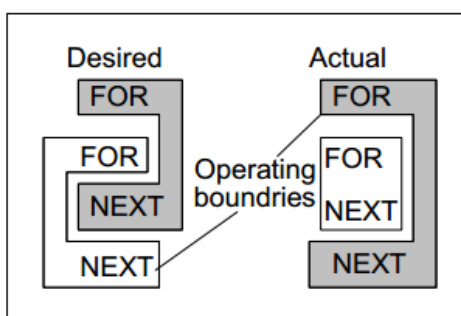


① Enters a state as if the DI instruction was missing. An error is not generated.

② The first occurrence of either an FEND or the END instruction takes priority.

This would then end the program scan prematurely.

③ The sequence will not process as expected, e.g.:



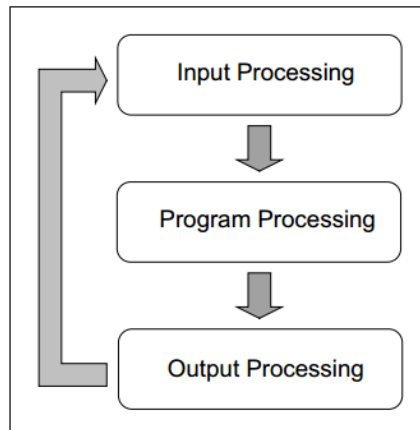
7.3 Batch Processing

This is the system used by all members of the HC family of PLC' s. The basic concept

is that there are three stages to any program scan. In other words, every time the

program is processed from start to end the following sequence of events occur.

Input processing:



All of the current input statuses are read in to a temporary memory area; sometimes called an image memory. The PLC is now ready for the next program processing.....

Program processing:

All of the updated inputs are checked as the

program is processed. If the new input statuses change the status of driven outputs, then these are

noted in the image memory for the.....

Output processing:

The new, current statuses of the outputs which have just be processed are physically updated, i.e relays are turned ON or OFF as required. The program scan starts again..... The system is known as 'Batch processing' because all of the inputs, program operation and finally the outputs are processed as batches.

7.4 Summary of Device Memory Allocations

The memory allocations of the programmable are very complex, but from a users point of view there are three main areas:

a) The Program Memory:

This memory area holds all of the data regarding: parameters, sequence program, constant values K and H, pointer information for P and I devices, nest level information, file register contents/allocations and also the program comment area.

- This memory area is latched either by battery backup or by use of EEPROM program management (dependent on the PLC being used). Any data stored in this area is kept even when the PLC is powered down. The duration and reliability of the data storage is dependent upon the condition of the battery or EEPROM being used to perform the backup process.

b) Data Memory

This memory area contains, as the title suggests, all of the data values associated with: data registers (normal and special), Index registers, current timer values, retentive timer values (if available) and current counter values.

- All of the devices which are designated as being latched (including retentive timers) are backed up in a similar method to the one mentioned under point a).

- Index registers and special data registers (D8000 to D8255) operate in the specified manner under the following circumstances.

Circumstance	Reaction
PLC's power is turned OFF	All data is cleared
PLC's power is turned ON	Certain devices are reset to their defaults see chapter 6
PLC is switched from STOP to RUN	Certain devices are reset to their defaults see chapter 6
PLC is switched from RUN to STOP	

- All other devices such as current values of non latched data registers, timers and counters behave in the following manner.

Circumstance	Reaction
PLC's power is turned OFF	All data is cleared
PLC's power is turned ON	
PLC is switched from STOP to RUN	No change
PLC is switched from RUN to STOP	Cleared (unless special M coil M8033 is active)

c) Bit Memory

This memory area contains the contact status of all inputs, outputs, auxiliary relays, state coils, timers and counters.

- All of the devices which are designated as being latched (including retentive timers) are backed up in a similar method to the one mentioned under point a).
- Special auxiliary relays (M8000 to M8255) act in a similar way to the special data registers mentioned under point b).
- All other devices are subject to the same changes as the current values of data registers, timers and counter (see the last point and table under section b).

Summary

Memory type	Power		PLC	
	OFF	OFF > ON	STOP > RUN	RUN > STOP
All devices backed by battery	Not changed			
Special M and D devices (8000 to 8255) and index registers V and Z	Cleared	Default	Not changed	
All other devices	Cleared		Not changed	Cleared
			Not changed when M8033 is set	

7.5 Limits Of Instruction Usage

7.5.1 Instructions Which Can Only Be Used Once In The Main Program Area

The following instructions can only be used once in the main program area. For PLC applicability please check either the detailed explanations of the instructions or the

instruction execution tables list earlier.



- Instructions which can only be used once are:

FNC 52 MTR	FNC 60 IST	FNC 70 TKY
FNC 57 PLSY	FNC 61 SORT	FNC 71 HKY
FNC 58 PWM	FNC 62 ABSD	FNC 72 DSW
FNC 59 PLSR	FNC 63 INCD	FNC 74 SEGL
FNC 68 ROTC	FNC 75 ARWS	



- Only one of either FNC 57 PLSY or FNC 59 PLSR can be programmed at once.

Both instructions can not be present in the same active program.

7.5.2 Instructions Which Are Not Suitable

For Use With 110V AC Input Units

When using 110V AC input units certain operations, functions and instructions are not recommended for use due to long energize/de-energize (ON/OFF) times of the 110V input devices.



- Program operations not recommended for use are:

- Interrupt routines

- High speed counters

- Instructions not recommended for use are:

FNC 51 REFF	FNC 68 ROTC	FNC 72 DSW
-------------	-------------	------------

FNC 52 MTR FNC 70 TKY FNC 75 ARWS

FNC 56 SPD FNC 71 HKY

8. PLC Device Tables

8.1 Performance Specification Of The HCA1P

Item	Specification	Remarks
Operation control method	Cyclic operation by stored program	
I/O control method	Batch processing method (when END instruction is executed	I/O refresh instruction is available
Operation processing time	Basic instructions: 0.55 to 0.7μs Applied instructions: 1.65 to several 100μs	
Programming language	Relay symbolic language + step ladder	Step ladder can be used to produce an SFC style program
Program capacity	2K steps	Provided by built in EEPROM memory
Number of instructions	Basic sequence instructions: 29 Step ladder instructions: 2 Applied instructions: 85	A Maximum 116 applied instructions are available including all variations
I/O configuration	Max total I/O set by Main Processing Unit	

Auxiliary relay(M coils)	General	384 points	M0 to M383
	Latched	128 points (subset)	M384 to M511
	Special	256 points	From the range M8000 to M8255
State relays (S coils)	General	128 points	S0 to S127
	Initial	10 points (subset)	S0 to S9
Timers (T)	100 msec	Range: 0 to 3,276.7 sec 63 points	T0 to T55
	10 msec	Range: 0 to 327.67 sec 31 points	T32 to T62 when special M coil M8028 is driven ON
	1msec	Range: 0.001 to 32.767 sec 1point	T63
Counters (C)	General	Range: 1 to 32,767 counts 16 points	C0 to C15 Type: 16 bit up counter
	Latched	16 points(subset)	C16 to C31 Type: 16 bit up counter
High speed counters (C)	1phase	Range: -2,147,483,648 to +2,147,483,647 counts	C235 to C240 (note C235 is latched) 6points
	1phase c/w start stop input		C241(latched), C242 and C244 (latched) 3 points
	2phase		C246, C247 and C249 (all latched) 3 points

	A/B phase		C251, C252 and C254 (all latched) 3 points
Data registers (D)	General	128 points	D0 to D127 Type:16 bit data storage register pair for 32 bit device
	Latched	128 points (subset)	D128 to D255 Type:16 bit data storage register pair for 32 bit device
	Externally adjusted	Range: 0 to 255 2 points	D8013 or D8030 & D8031 Data is entered indirectly through the external setting potentiometer
	Special	256 points (inclusive of D8013)	From the range D8000 to D8255 Type: 16 bit data storage register
	Index	16 points	VandZ Type: 16 bit data storage register
Pointers (P)	For use with CALL	64 points	P0 to P63
	For use with interrupts	6 points	I00□ to I30□ (rising trigger□=1,

			falling trigger□=0)
Nest levels		8 points for use with MC and MCR	N0 to N7
Constants	Decimal K	16 bit: -32,768 to +32,767 32 bit: -2,147,483,648 to +2,147,483,647	
	Hexadecimal	16 bit: 0000 to FFFF	
	H	32 bit: 00000000 to FFFFFFFF	

8.2 Performance Specification Of The HCA2P/HCA2C

Item	Specification	Remarks
Operation control method	Cyclic operation by stored program	
I/O control method	Batch processing method (when END instruction is executed)	I/O refresh instruction is available
Operation processing time	Basic instructions: 0.55 to 0.7μs Applied instructions: 1.65 to several 100μs	
Programming language	Relay symbolic language + step ladder	Step ladder can be used to produce an SFC style program
Program capacity	8K steps	Provided by built in EEPROM memory
Number of instructions	Basic sequence instructions: 29	A Maximum 120 applied instructions are available including all variations

		Step ladder instructions: 2	
		Applied instructions: 89	
I/O configuration		Max hardware I/O configuration points 128, dependent on user selection (Max. software addressable Inputs 128, Outputs 128)	
Auxiliary relay (M coils)	Genera	384 points	M0 to M383
	Latched	1152 points (subset)	M384 to M1535
	Special	256 points	From the range M8000 to M8255
State relays (S coils)	Latched	1000 points	S0 to S999
	Initial	10 points (subset)	S0 to S9
Timers (T)	100 msec	Range: 0 to 3,276.7 sec 200 points	T0 to T199
	10 msec	Range: 0 to 327.67 sec 46 points	T200 to T245
	1msec	Range: 0 to 32.767 sec 4point	T246 to T249
	100 msec retentive	Range: 0 to 3,276.7 sec 6 points	T250 to T255
Counters (C)	General	Range: 1 to 32,767 counts 16 points	C0 to C15 Type: 16 bit up counter
	Latched	184 points (subset)	C16toC199 Type: 16 bit up counter

	General	Range: 1 to 32,767 counts 20 points	C200 to C219 Type: 32 bit bi-directional counter
	Latched	15 points (subset)	C220 to C234 Type: 32 bit bi-directional counter
High speed counters (C)	1phase	Range: -2,147,483,648 to	C235 to C240 6points
	1phase c/w start stop input	+2,147,483,647 counts Select upto four 1 phase counters with a combined counting frequency of 5kHz or less. Alternatively select one 2 phase or A/B phase counter with a counting frequency of 2kHz or less. Note all counters are latched	C241, C242 and C244 3points
	2phase		C246, C247 and C249 3points
	A/B phase		C251, C252 and C254 3points
Data registers (D)	General	7128 points	D0 to D127 & D1000 to D7999 Type: 16 bit data storage register pair for 32 bit device
	Latched	872 points (subset)	D128 to D999 Type: 16 bit data storage register pair for 32 bit device
	File	7000 points	D1000 to D6999 set by parameter in 3

			blocks of 500 program steps Type: 16 bit data storage register
	Externally adjusted	Range: 0 to 255 2 points	Data is move from external setting potentiometers to registers D8030 and D8031)
	Special	256 points (inclusive of D8013, D8030 and D8031)	From the range D8000 to D8255 Type: 16 bit data storage register
	Index	16 points	VandZ Type: 16 bit data storage register
Pointers (P)	For use with CALL	128 points	P0 to P127
	For use with interrupts	6 points	I00□ to I30□ (rising trigger□=1, falling trigger□=0)
Nest levels		8 points for use with MC and MCR	N0 to N7
Constants	Decimal K	16 bit: -32,768 to +32,767 32 bit: -2,147,483,648 to +2,147,483,647	
	Hexadecimal	16 bit: 0000 to FFFF	
	H	32 bit: 00000000 to FFFFFFFF	

9. Assigning System Devices

9.1 Addressing Extension Modules

Most of the HC family of PLC's have the ability to connect additional discrete I/O and/or special function modules. To benefit from these additional units the user must address each block independently.

Addressing Additional Discrete I/O

This type of I/O is the standard input and output modules. As each extension block or powered extension unit is added to the system they assume the next available addresses. Hence, the units closest to the base unit will have the lowest I/O numbers or addresses. I/O numbers are always counted in octal. This means from 0 to 7 and 10 to 17 etc. Within a users program the additional addresses are used as normal.

Discrete I/O can be added at the users discretion as long as the rules of system configuration for each PLC type are obeyed. This information can be found in the appropriate hardware manual. For easy use and identification, each additional I/O unit should be labeled with the appropriate I/O numbers using the provided number labels.



Caution when using an system with 8ER, 24MR units

- When an 8ER or an 24MR are used an additional 8 points (as 4 inputs, 4 outputs) of

I/O must be allowed for. This is because both units split blocks of 8 inputs and 8 outputs to obtain a physical 4 input/ 4 output configuration. Hence, an 8ER unit actually occupies 8 input points and 8 output points even though there are only 4 physical inputs and 4 physical outputs.

Addressing Special Function Blocks

Special function blocks are allocated a logical ‘station/block number’ from 0 to 7. This is used by the FROM/TO instructions to directly access each independent special function module.

The lower the ‘station/block number’ is, the closer to the base unit it can be found.

Special function blocks can be added at the users discretion but the rules of configuration for each type of PLC must be obeyed at all times. The configuration notes can be found in the appropriate hardware manual for each programmable controller.

9.2 Real Time Clock Function

The time data of a RTC cassette or chip (built in to HCA1P and HCA2P/HCA2C) is battery backed. This means when the PLC is turned OFF the time data and settings are not lost or corrupted. The duration or storage life of the timedetails dependent upon the condition of the battery.

The real time clock has a worst case accuracy of ± 45 seconds per month at an ambient temperature of 25°C. The calendar function of the RTC caters for leap years

during the period 1980 through 2079.

9.2.1 Setting the real time clock

The RTC can be set using the special data registers and control flags as follows:

			Device Number	Comments
			M8015 Time setting	Set ON to stop the clock. When the clock is stopped the time values can be reset. The clock restarts when the flag is reset to OFF.
D8013	Seconds	0 to 59	M8016 Register Hold	The clock data in the data registers is held. The clock still runs. Use this to pause the data to read the current time.
D8014	Minutes	0 to 59		
D8015	Hours	0 to 23		
D8016	Date	1 to 31 (correct for current Month)	M8017 Minute Rounding	When on rounds the time up or down to the nearest minute.
D8017	Month	1 to 12	M8018 Clock Available	Automatically set to indicate the RTC is available.
D8018	Year	00 to 99 (1980 to 2079)		
D8019	Day of Week	0 to 6 (Sunday to Saturday)	M8019 Setting Error	ON when the values for the RTC are out of range.

9.3 Analog Expansion Boards

The HCA2P/HCA2C expansion boards can be installed on the HCA1P Series PLCs to provide extra analog I/O channels. Please see the respective expansion board User's Manual for more information on configuration and hardware specifications.

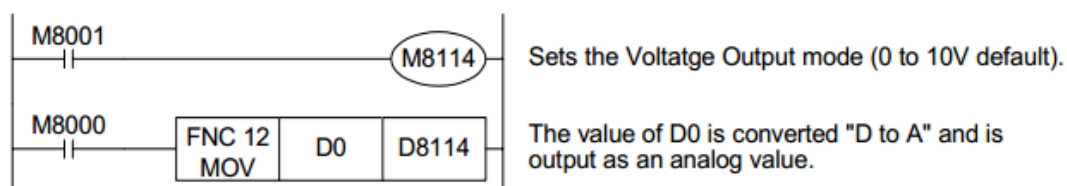
The expansion boards are not equipped with a Gain/Offset setting so that these values must be calculated in the PLC ladder program. Example programs are provided below.

9.3.1 TX1N-1DA-BD

This expansion board is used to convert a digital value in the range of 0 ~ 4000 that is stored in D8114 to an analog output value. The analog output can be in the Voltage range of 0-10 Volts DC or 4-20mA.

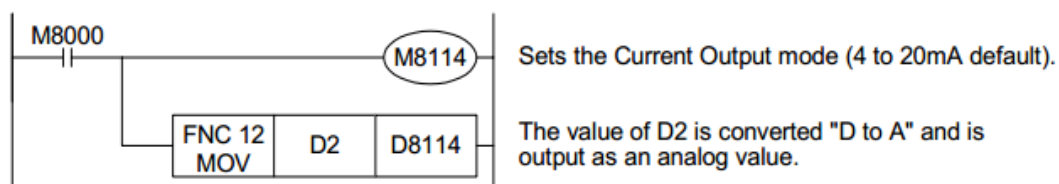
Voltage Output Mode

The following program example sets the Voltage Output mode. A digital value in D0 is converted to the analog equivalent for output.



Current Output Mode

The following program example sets the Current Output mode. A digital value in D0 is converted to the analog equivalent for output.



Example Application Programs

The user can use any digital value range that is convenient in the program but must convert the value to the 0 ~ 4000 range before the correct analog value can be output.

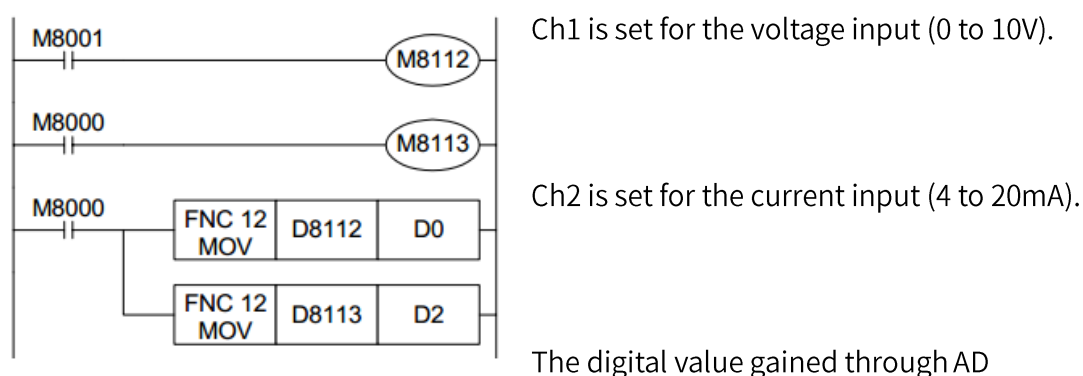
In the same way, the analog outputs can be modified via PLC programming to give

outputs within a certain range. Please note that outputs outside the given range are not possible.

The Please see programming examples below.

Example Application Program #1

Output an analog value in the range of 0 to 10 Volts when the digital value in the user program is 0 ~ 10000.

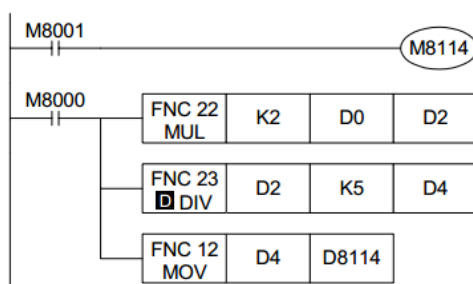


conversion of Ch1 is stored at D0.

The digital value gained through AD conversion of Ch2 is stored at D2

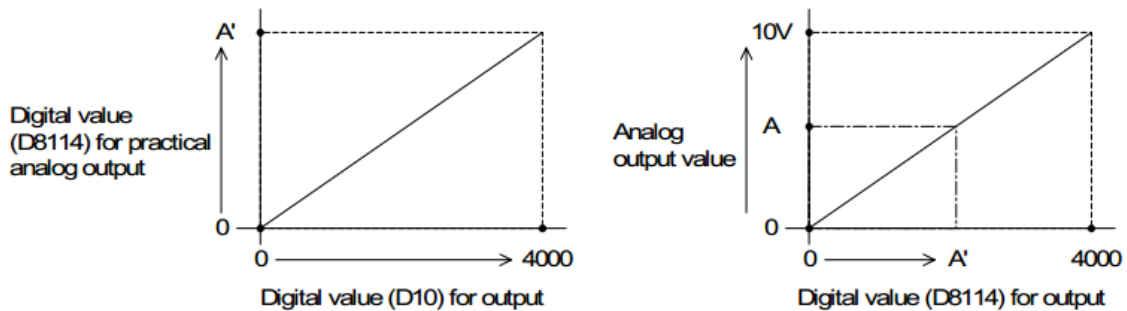
D0 ranges from 0 ~10000. To convert D0 to the 0 ~ 4000 value needed for D8114:

$$D8114 = [D0 \times 4000] / 10000 \text{ or } [D \times 2] / 5$$



Example Application Program #2

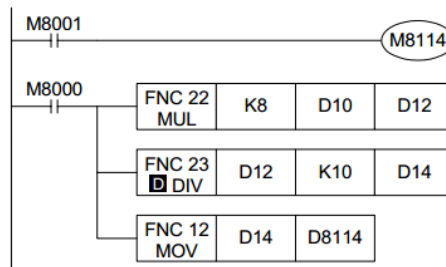
An output of $0 \sim A$ [$0 < A < 10$] is desired in the program that is using a digital range of $0 \sim 4000$ that is stored in register D10.



Because A is smaller than 10 Volts, the digital value of $0 \sim 4000$ must be converted to a value of $0 \sim A'$ as shown in the graphs above. $4000/10V = A' / A$ or $A' = [4000/10] \times A = 400 \times A$

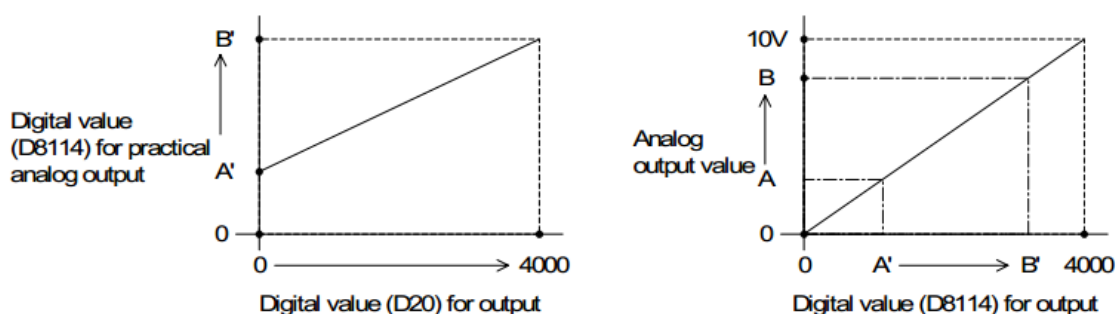
$$D8114 = [A'] \times (D10 / 4000) = [400 \times A] \times [D10 / 4000] = (A \times D10) / 10.$$

If $A = 8$



Example Application Program #3

The desired analog output is from values A to B where $0 < A < B < 10$ and the digital values range from $0 \sim 4000$ in D20.



This example is equivalent to setting an offset and gain for the analog output.

The digital values must be converted to A' and B' per the graphs above.

$$[B-A]/[10-0]=[B'-A'] / [4000 - 0], \text{ therefore } [B' - A'] = [B - A] \times 400.$$

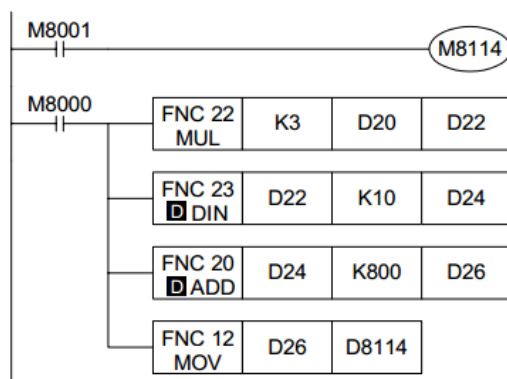
$$D8114 = [B' - A'] \times (D20 / 4000) + A'$$

$$B' = 400 \times B \text{ and } A' = 400 \times A \text{ (see previous example programs for calculation)}$$

$$D8114 = [400 \times (B - A)/4000] \times D20 + (400 \times A)$$

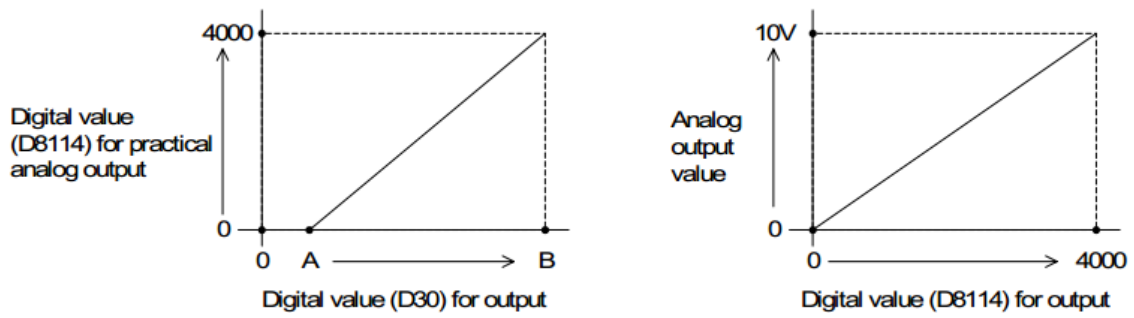
$$D8114 = [(B-A)/10] \times D20 + (400 \times A)$$

If A = 2 and B = 5, see the programming example below



Example Application Program #4

In Voltage Output Mode, a digital range of values A ~ B is used in the program for an analog output of 0 ~ 10 Volts. The digital range of A ~ B stored in D30 must be converted to 0 ~ 4000 before the correct analog value can be output.

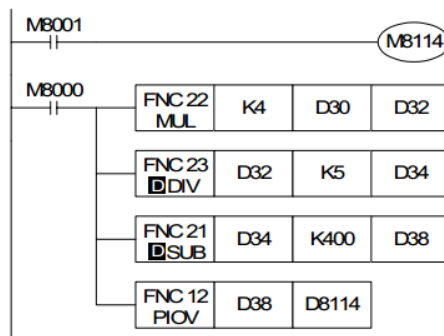


$$[(4000 - 0) / (B - A)] = D8114 / (D30 - A)$$

$$D8114 = [4000 \times D30 / (B - A)] - [(4000 \times A) / (B - A)]$$

If A = 500 and B = 5500, then

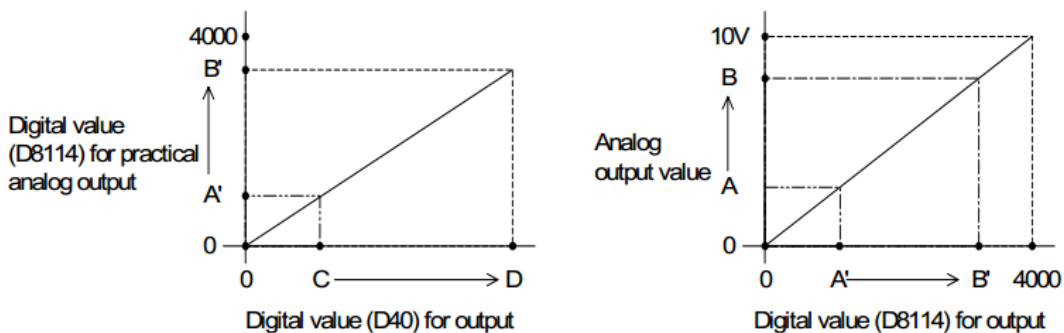
$$D8114 = (4/5) \times D30 - 400$$



Example Application Program #5

If using a digital range of C ~ D in the program to output an analog value of A ~ B, the digital value must be converted to the 0 ~ 4000 equivalent and the analog value must be converted to 0 ~ 10 Volt equivalent.

Digital Values for conversion to analog are stored in D8114



Please see prior programming examples for sample equations for the conversion of data ranges.

$$D8114 = [(B' - A') \times D40] / (D - C) + [(A' \times D) - (B' \times C)] / (D - C)$$

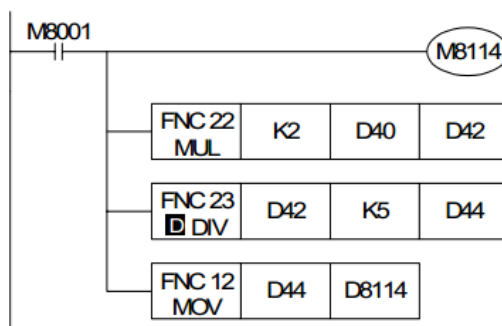
$$D8114 = [(400 \times B - 400 \times A) \times D20] / (D - C) + [(400 \times A \times D) - (400 \times B \times C)] / (D - C) \text{ (from prior)}$$

examples $A' = 400 \times A$ and $B' = 400 \times B$

$$D8114 = [400 \times (B - A)] / (D - C) + 400 \times [(A \times D) - (B \times C)] / (D - C)$$

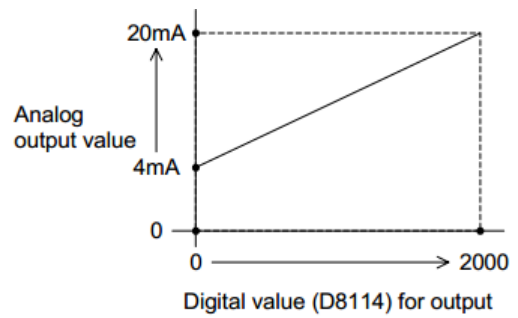
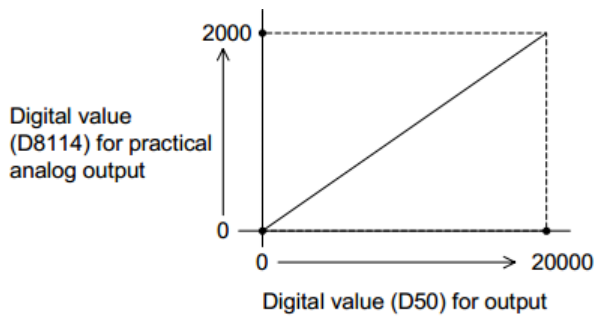
If $A = 1$, $B = 5.5$, $C = 1000$, and $D = 5500$, then

$$D8114 = (2 \times D40) / 5$$



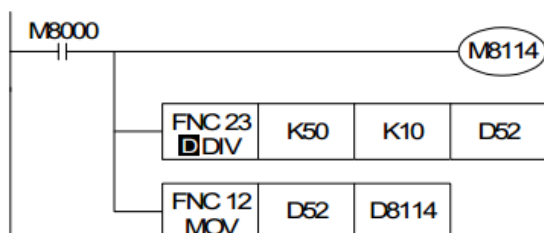
Example Application Program #6

In the Current Output Mode, the 1DA converts values from 0 ~ 2000 to the analog output of 4 ~ 20 mA. If using a digital range of 0 ~ 20000 in the program, the range must be converted to 0 ~ 2000 as shown in the programming example below. Digital values for conversion to analog are stored in D8114.



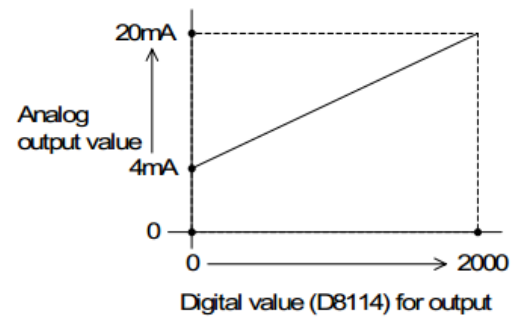
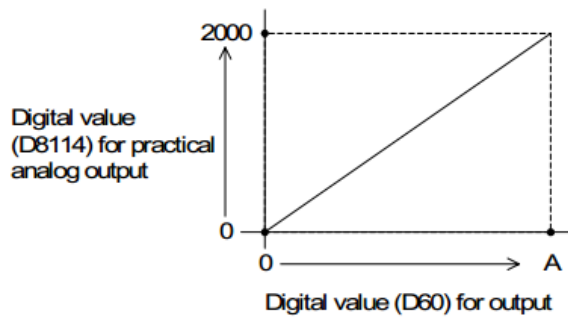
$$D8114 = [(2000 - 0) \times D50] / (20000 - 0)$$

$$D8114 = D50 / 10$$



Example Application Program #7

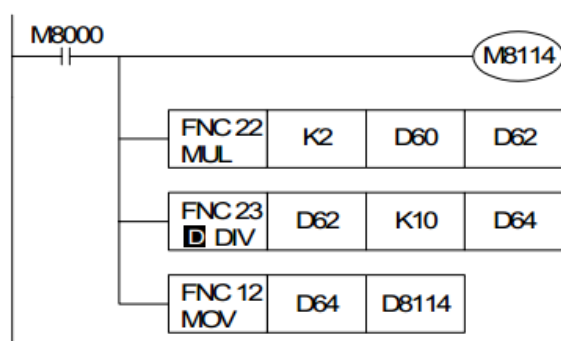
In Current Output Mode, a user wants to use a range of 0 ~ A in the program to output the analog current of 4 ~ 20mA. The user range 0 ~ A stored in D60 must be converted to the range of 0 ~ 2000 as shown below.



$$D8114 = [(2000 - 0) \times D60] / (A - 0)$$

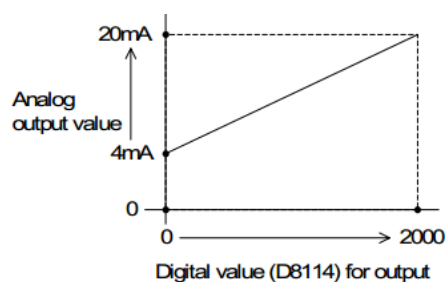
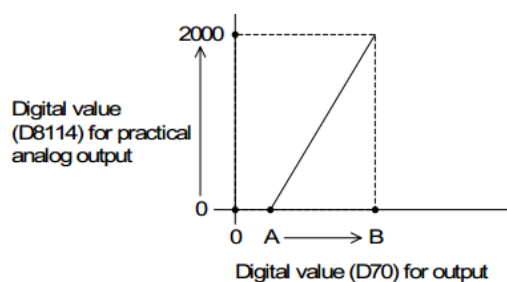
$$D8114 = (2000 \times D60) / A, \text{ if } A = 10000$$

$$D8114 = D60 / 5$$



Example Application Program #8

In Current Output mode, the user digital range of A ~ B is used to output a current of 4 - 20 mA. The range of A ~ B stored in D70 must be converted to a range of 0 ~ 2000 per the example program below.

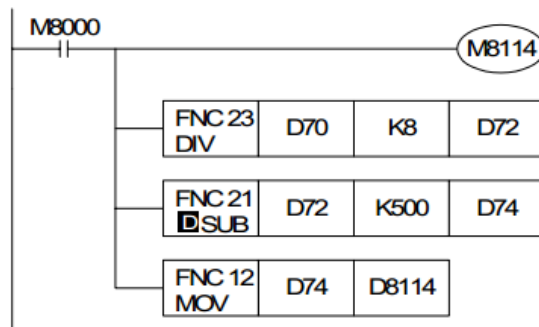


$$D8114 / (D70 - A) = (2000 - 0) / (B - A)$$

$$D8114 = \{ [(2000 - 0) \times D70] / (B - A) \} - \{ [(2000 - 0) \times A] / (B - A) \}$$

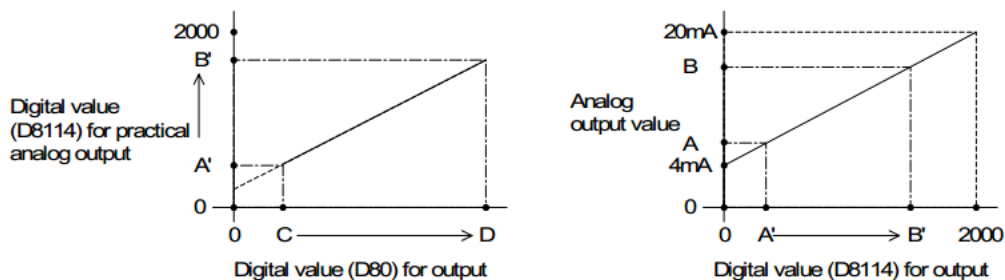
If A = 4000 and B = 20000, then $[(2000 \times D70) / (20000 - 4000)] - [2000 \times 4000 / (20000 - 4000)]$

$$D8114 = (D70 / 8) - 500$$



Example Application Program #9

In Current Output mode, a current in the range of A ~ B (4mA < A < B < 20 mA) is output by using a digital range of C ~ D that is stored in D80. The current range A ~ B must be converted to the 4 ~ 20mA equivalent value and the digital range C ~ D must be converted to the 0 ~ 2000 range equivalent value.



Please see previous programming examples for sample range conversion calculations.

$$D8114 = (B' - A') \times D80 / (D - C) + \{(A' \times D) - (B' \times C)\} / (D - C)$$

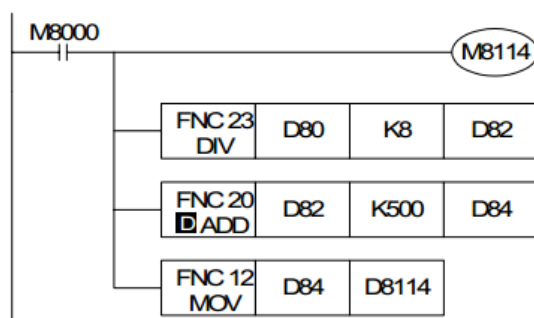
$$A' = 125 \times A - 500, B' = 125 \times B - 500,$$

$$D8114 = [(125 \times B - 500) - (125 \times A - 500)] \times D80 / (D - C) + [(125 \times A - 500) \times D - (125 \times B - 500) \times C] / (D - C)$$

If A = 5, B = 15, C = 5000, and D = 15000

$$= [125 \times (15 - 5)] \times D80 / (15000 - 5000) + 125 \times [(5 - 4) \times 15000 - (15 - 4) \times 5000] / (15000 - 5000)$$

$D8114 = (D80/8) - 500$

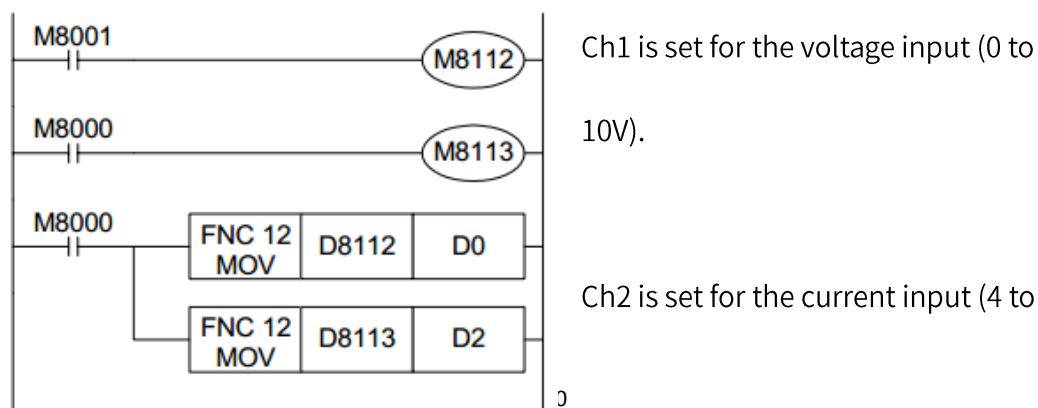


9.3.2 TX1N-2AD-BD

This expansion board is used to convert up to two channels of analog input into digital values for use by the HCA1P Series PLCs. Voltage input (0 ~ 10 Volts) or Current input (4 to 20 mA) for analog to digital conversion can be set by switching the auxiliary relays assigned to each channel. The output values can be adjusted after the conversion via PLC program code but resolution cannot be improved.

Basic Program #1

The following program sets Channel 1 in the Voltage Input mode and Channel 2 in the Current Input mode with the A/D converted digital value of each channel stored in D0 and D2 respectively.



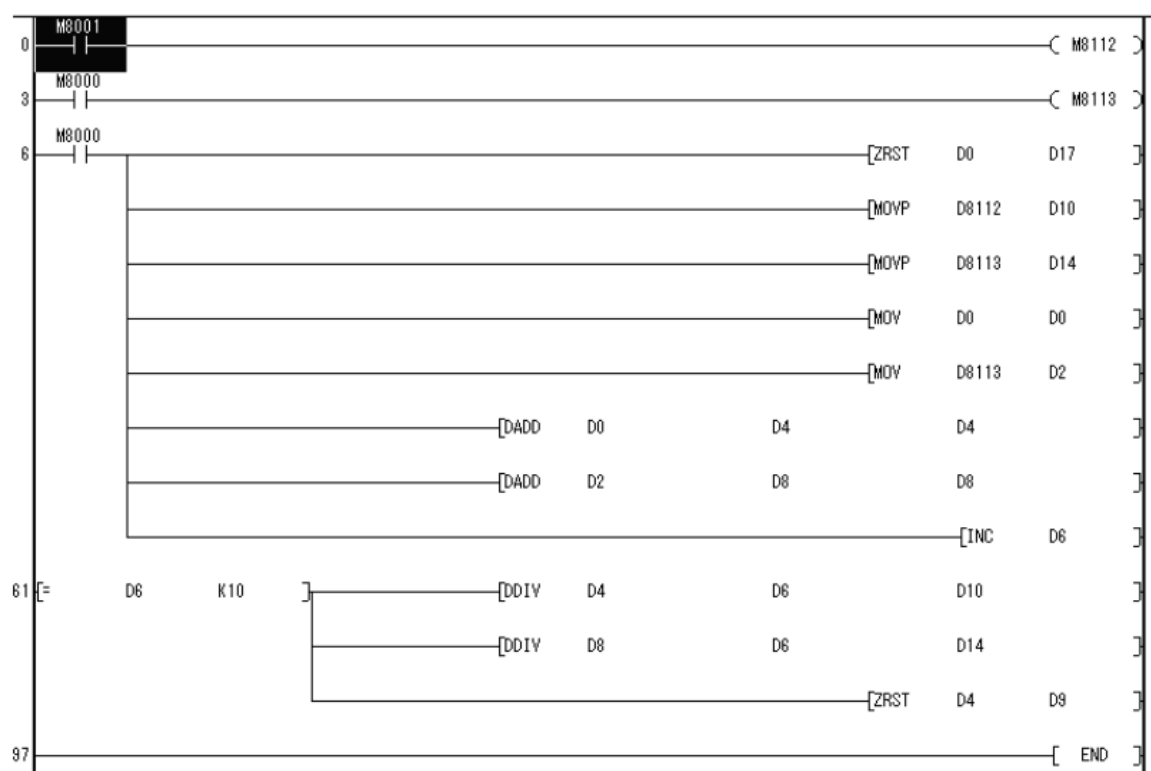
20mA).

The digital value gained through AD conversion of Ch1 is stored at D0.

The digital value gained through AD conversion of Ch2 is stored at D2.

Basic Program #2

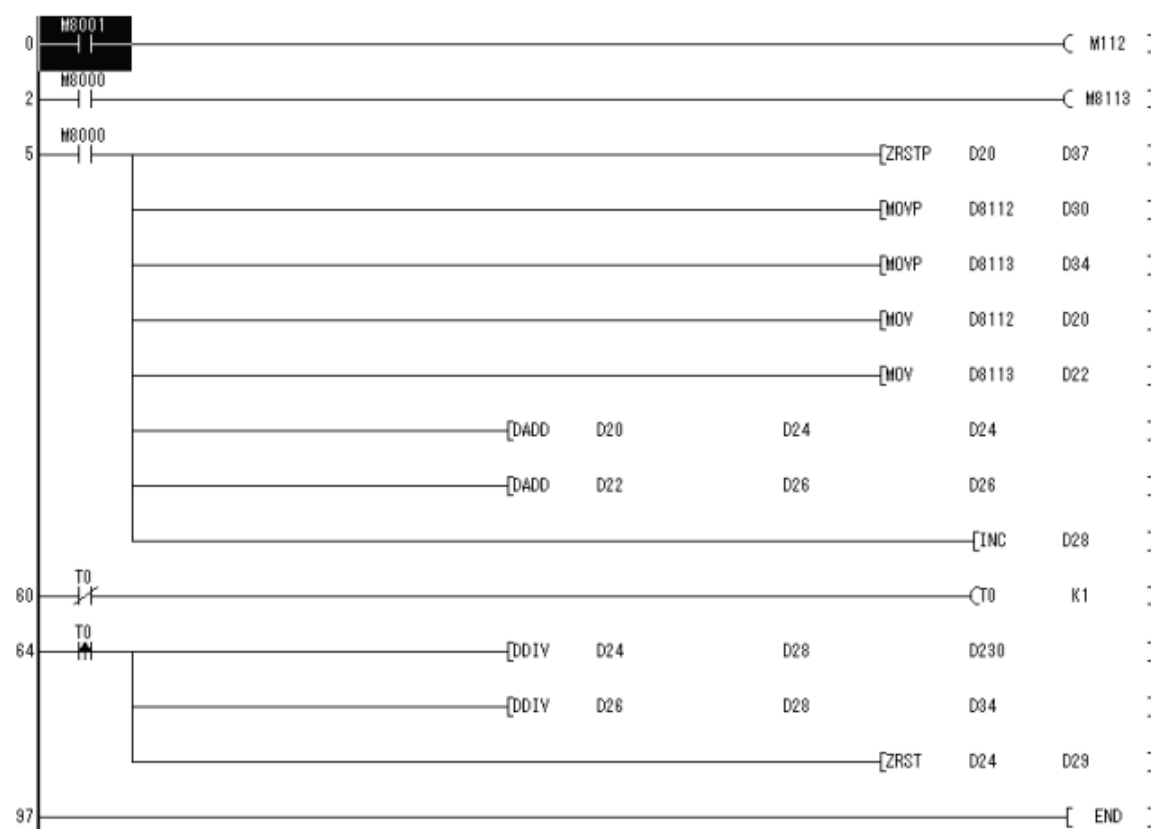
Ch1 is set to Current input, Ch2 is set to Voltage input, and the average converted digital value over a set time period is stored in D10 and D14.



Basic Program 3

Ch1 is set to Current input, Ch2 is set to Voltage input, and the average converted

digital value over a set time period is stored in D30 and D34, respectively.



Example Application Programs

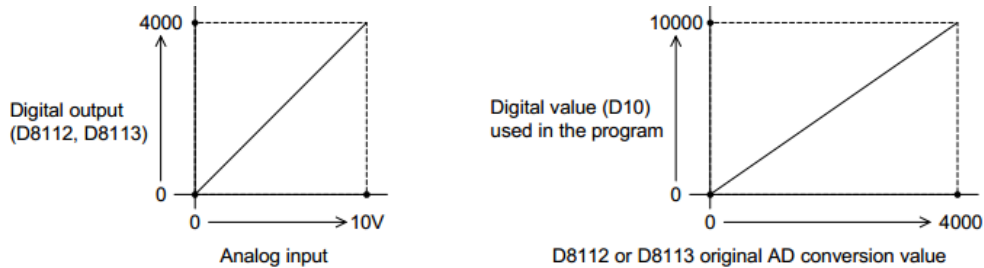
Because the 2AD does not have Offset and Gain capabilities, if values are required outside the standard specification range, additional program commands are required to either multiply or divide the conversion values.

When adjusting the conversion values, some of the resolution will be lost. The original range of the analog input does not change.

Example Application Program #1

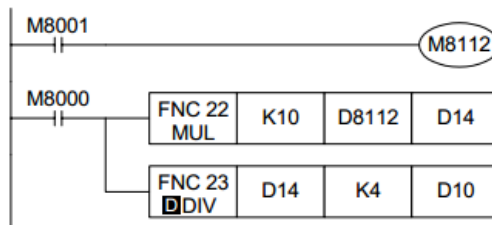
In Voltage input mode, the 2AD converts analog values from 0 ~ 10 Volts to a digital output of 0 ~ 4000. If using a digital range of 0 ~ 10000 in the program, the 0 ~ 4000

output value must be converted as shown in the programming example below. Digital values that are converted from analog values are stored in D8112 or D8113.



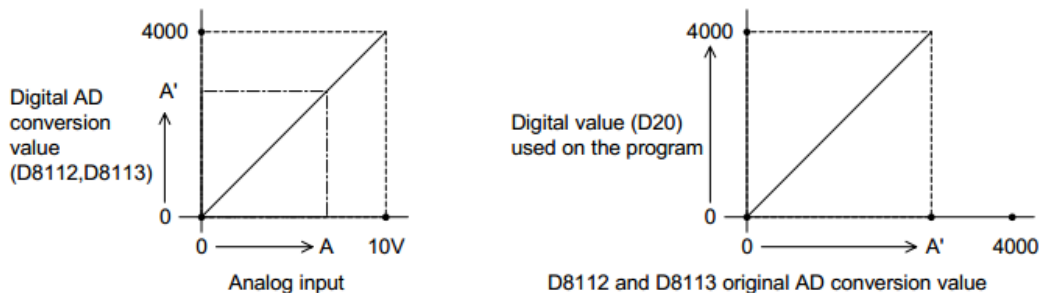
$$D10 = 10 \times D8112 / 4, \text{ (D8113 would be used for Ch2)}$$

The programming code for the Equation above is given below.



Example Application Program #2

In Voltage input mode, the 2AD converts analog values from 0 ~ 10 Volts to a digital output of 0 ~ 4000. If using an analog range of 0 ~ A (where 0 < A < 10) by a digital output range of 0 ~ 4000, the range must be converted from 0 ~ A' to 0 ~ 4000 as shown in the programming code below.



If a digital value of 0 ~ 4000 is used in D20,

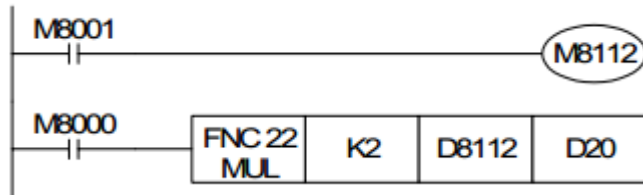
$$D20 = (4000) \times (D8112 \text{ or } D8113) / A'$$

$4000 / (10 \text{ volts}) = A' / (A \text{ volts})$, therefore $A' = 400 \times A$

$D20 = 4000 \times (D8112 \text{ or } D8113) / 400 \times A$

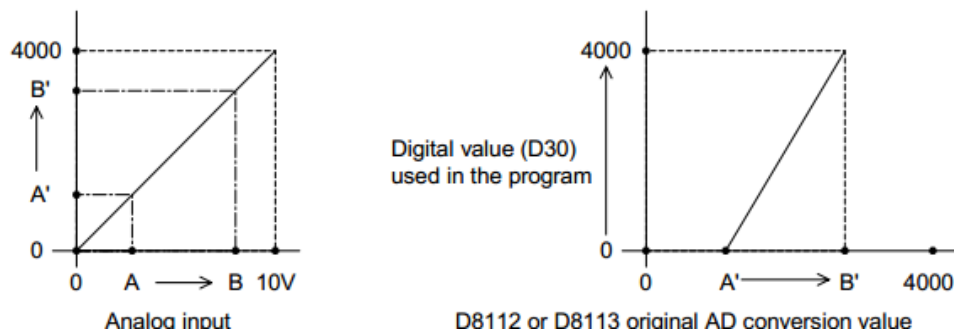
$D20 = 10 \times (D8112 \text{ or } D8113) / A$ and if $A=5$

$D20 = 2 \times (D8112 \text{ or } D8113)$



Example Application Program #3

If using an analog range from A ~ B by a digital range of 0 ~ 4000, the range must be converted from A' ~ B' 0 ~ 4000 in the program as shown in the example below.



If the digital range 0 ~ 4000 is desired in D30, please see the program below.

$D30 = 4000 \times (D8112 \text{ or } D8113) / (B' - A') - 4000 \times A' / (B' - A')$

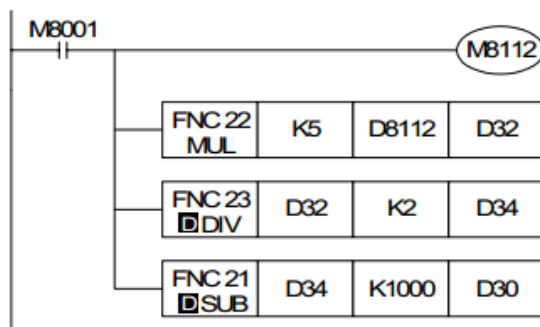
$A' = 400 \times A, B' = 400 \times B$ so that

$D30 = [4000 \times (D8112 \text{ or } D8113) / (400 \times B - 400 \times A)] - 4000 \times (400 \times A) / (400 \times B - 400 \times A)$

$D30 = [10 \times (D8112 \text{ or } D8113) / (B - A)] - 4000 \times A / (B - A)$

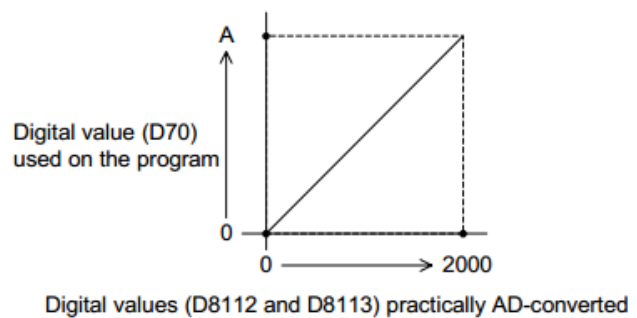
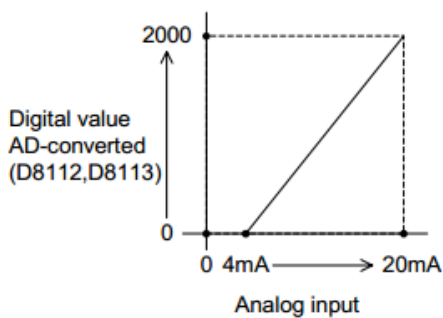
If $A=1$ and $B=5$

$D30 = [5 \times (D8112 \text{ or } D8113) / 2] - 1000$



Example Application Program #4

If using an analog range from 4 ~ 20mA to obtain an output range from 0 to A, the normal output range of 0 ~ 2000 be converted to the new range.

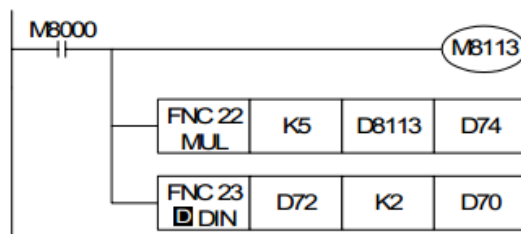


Please perform the conversion as below.

$D70 = A \times (D8112 \text{ or } D8113) / 2000$. If $A = 5000$ then,

$D70 = 5000 \times (D8112 \text{ or } D8113) / 2000$

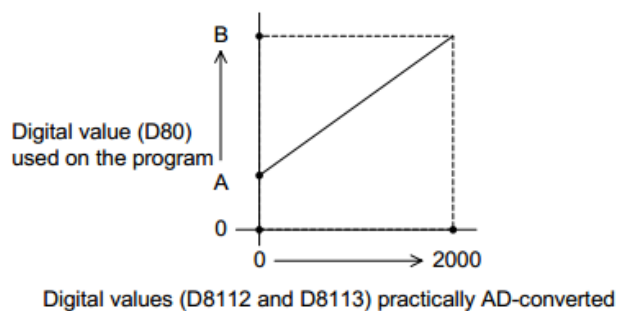
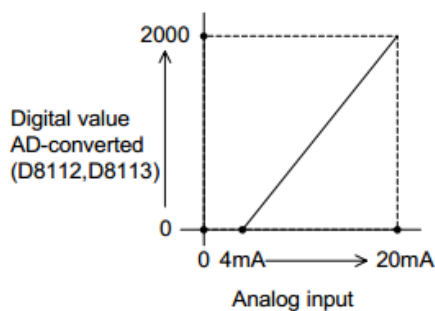
$D70 = 5 \times (D8112 \text{ or } D8113) / 2$



Example Application Program #5

If using an analog range from 4 ~ 20mA to obtain an output range from A ~ B, the

normal output range of 0 ~ 2000 must be converted to the new range.

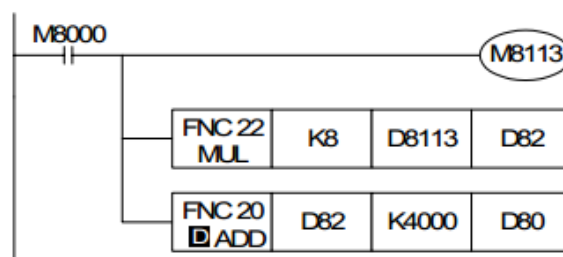


To convert the normal output range of 0 ~ 2000 to the range of A ~ B, please see below.

$$D80 = (B - A) \times (D8112 \text{ or } D8113) / (2000 - 0) + A; \text{ if } A = 4000 \text{ and } B = 20000$$

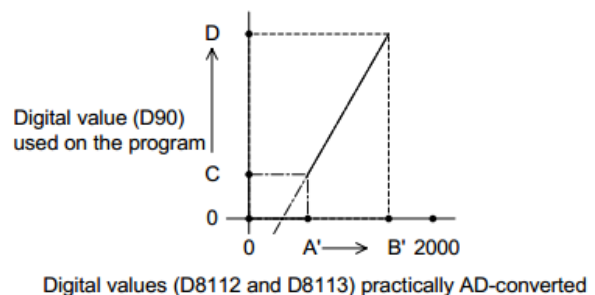
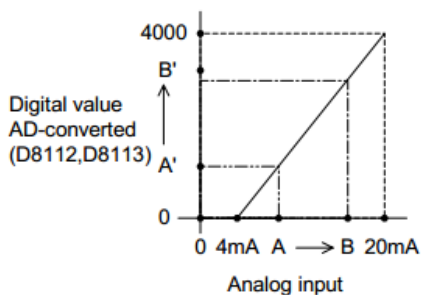
$$D80 = (20000 - 4000) \times (D8112 \text{ or } D8113) / (2000) + 4000$$

$$D80 = 8 \times (D8112 \text{ or } D8113) + 4000$$



Example Application Program #6

If using an analog range from A ~ B to obtain an output range from C ~ D, both the current and the digital ranges must be converted from the standard ranges.



To convert both ranges, please see the programming example below. More details can

be found from the previous examples.

$$D90 = (D - C) \times (D8112 \text{ or } D8113) / (B' - A') + (B' \times C - A' \times D) / (B' - A')$$

$$D90 = (D - C) \times (D8112 \text{ or } D8113) / [(125 \times B - 500) - (125 \times A - 500)] + [(125 \times B - 500) \times C - (125 \times A - 500) \times D] / [(125 \times B - 500) - (125 \times A - 500)]$$

$$(A' = 125 \times A - 500; B' = 125 \times B - 500)$$

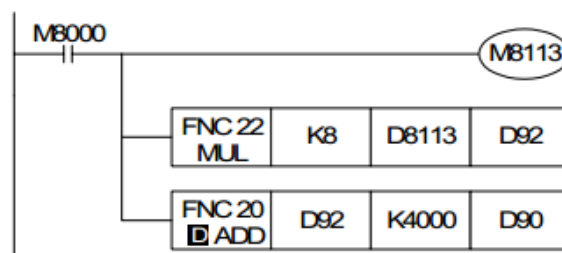
$$D90 = (D - C) \times (D8112 \text{ or } D8113) / [125 \times (B - A)] + [(B - A) \times C - (A - 4) \times D] / (B - A)$$

If A = 5, B = 15, C = 5000, and D = 15000

$$D90 = (15000 - 5000) \times (D8112 \text{ or } D8113) / [125 \times (15 - 5)] +$$

$$[(15 - 4) \times 5000 - (5 - 4) \times 15000] / (15 - 5)$$

$$D90 = 8 \times (D8112 \text{ or } D8113) + 4000$$



10. Points Of Technique

10.1 Advanced Programming Points

The HC family of programmable controllers has a very easy to learn, easy to use instruction set which enables simple programs to perform complex functions. This chapter will point out one or two useful techniques while also providing the user with valuable reference programs



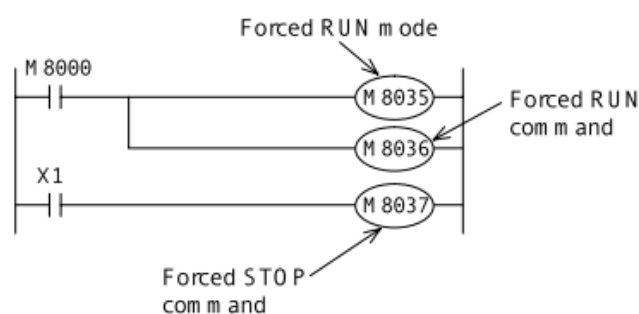
If some of these techniques are applied to user programs the user must ensure that they will perform the task or operation that they require. HCFA Corporation can take no responsibility for user programs containing any of the examples within this manual.

Each program will include a brief explanation of the system. Please note that the method of 'how to program' and 'what parameters are available' for each instruction will not be discussed. For this information please see the relevant, previous chapters.

10.2 Using The Forced RUN/STOP Flags

10.2.1 A RUN/STOP push button configuration

The HCFA programmable controller has a single RUN terminal. When power is applied to this terminal the PLC changes into a RUN state, i.e. the program contained is executed. Consequently when there is no power 'on' the RUN terminal the PLC is in a STOP state. This feature can be utilized to provide the HC PLC with an external RUN/STOP - push button control. The following PLC wiring and program addition are required.



Explanation:

Pressing the RUN push button sets the PLC into the RUN state. This means M8000 is ON.

Following the program, M8000 activates both M8035 and M8036. These two special auxiliary devices set the PLC in to forced RUN mode. Releasing the RUN push button would normally return the PLC to the STOP state, but because the two auxiliary coils, M8035 and 36 are ON, the PLC remains in RUN. To stop the, PLC pressing the STOP push button drives an input ON and consequently M8037 turns ON. This then automatically forces OFF both M8035 and 36 and resets itself. Hence, the PLC is in its STOP status and awaits the cycle to begin again.

Input priority:

- The STOP input is only processed after the programs END statement has been reached -this is because the physical input used, i.e. an X device is normally updated and processed at that time. Therefore, the RUN input is given priority when both RUN and STOP inputs are given simultaneously.
- To give priority to the STOP input and provide a 'safer' system, some form of mechanical/ circuitry interlock should be constructed between both RUN and STOP inputs. A very simple example is shown in the wiring diagram above.

10.2.2 Remote RUN/STOP control

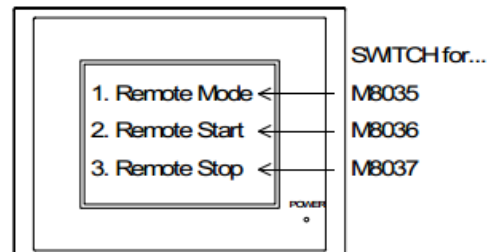
The HCFA programmable controllers can be controlled, i.e. switched into RUN or STOP modes and have devices monitored by use of intelligent external control devices.

These includes such items as computers, the HCFA data access units and Graphic Operator Terminals.

The following example utilizes a graphic unit:

Explanation:

The programmable controller needs no special wiring or additional programming for this example.



The only condition required is that the PLC would not normally be in a RUN state, i.e., there is no connection to the RUN terminal and the RUN/STOP switch on PLC's that have one is set in the STOP position.

The HMI should be programmed with 'SWITCH' devices driving the three special M codes M8035,36 and 37. By activating the 'SWITCH' devices for M8035 and M8036 the

SWITCH	
INPUT:	0
OUTPUT: PLC M8035	
MODE: ALTERNATE	

PLC can be switched into a RUN state, while driving the 'SWITCH' device M8037 will put the PLC into a STOP state.

Example 'SWITCH' device setting opposite.

Use an 'Alternate' switch for M8035 and M8036 and use a 'Momentary' switch for M8037. (see DU operation manual for SWITCH operation and programming)



Note: While M8035 and M8036 are ON the MPU can not be changed to STOP mode using the RUN terminal or RUN/STOP switch. Either set M8037 ON, or reset M8035 and M8036, to return to the normal operating state.

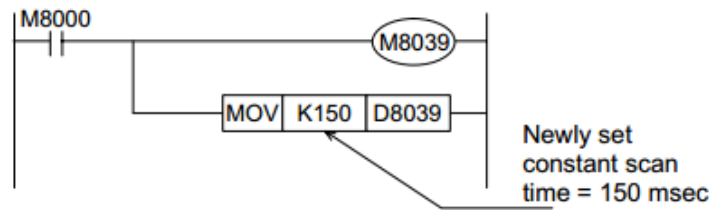
HCA2P,HCA2C Remote STOP

With HCA2P,HCA2C units, even if the RUN terminal or RUN/STOP switch is in the RUN position, it is still possible to do a remote STOP by forcing M8037 ON. Return to RUN by resetting M8037.

10.3 Constant Scan Mode

Some times the timing of operations can be a problem, especially if some co-ordination is being attempted with a second control system. In cases like this it is very useful to fix the PLC' s scan time. Under normal conditions the PLC' sscantimewillvaryfromonescantothe next. This is simply because the natural PLC scan time is dependent on the number of and type of the active instructions. As these are continually changing between program scans the actual scan time is also a varying. Hence, by using the additional program function identified below, the PLC' s scan time can be fixed so that it will be the same duration on every program scan. The actual scan duration is set by writing a scan time in excess of the current longest scan duration to special data register D8039 (in the example the value K150 is used). If the PLC scans the program quicker than the set scan time, a 'pause' will occur until the set scan duration is reached.

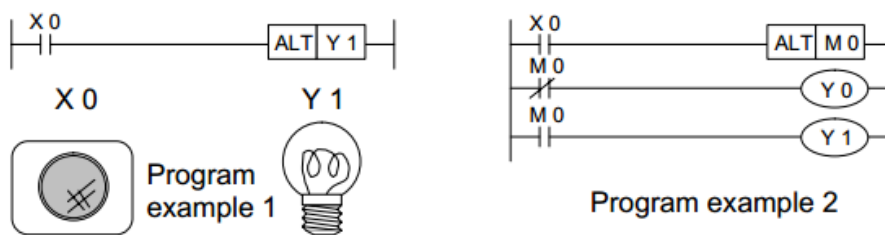
This program example should be placed at the beginning of a users program.



10.4 Alternating ON/OFF States

It is often useful to have a single input control or toggle a situation. A basic, yet typical example is the switching ON/OFF of a Light. This can be easily achieved by using standard ladder program to load an input and switch an output. However, this system requires an input which is latchable. If basic ladder steps are used to latch the program then it soon becomes complex and prone to mis-programming by the user. Using the ALT instruction to toggle the ON/OFF (SET/RESET, START/STOP, SLOW/FAST) state is much simpler, quicker and more efficient.

Explanation:



Pressing the momentary push button X1 once will switch the lamp ON. Pressing the push button for a second time will cause the lamp to turn OFF. And if the push button is again pressed for a third time, the lamp is turned ON again and so the toggled status

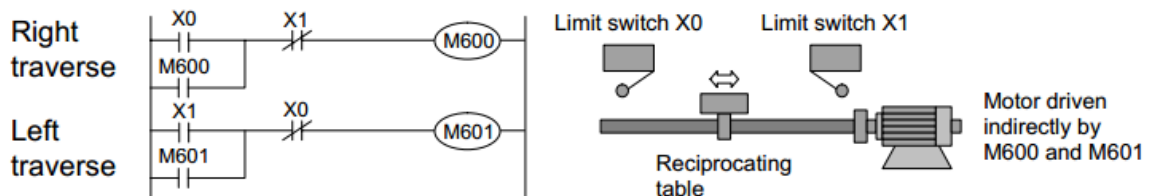
continues. The second program shown identifies a possible motor interlock/control, possibly a start/stop situation.

10.5 Using Battery Backed Devices

For Maximum Advantage Battery backed devices retain their status during a PLC power down. These devices can be used for maximum advantage by allowing the PLC to continue from its last operation status just before the power failure.

For example: A table traverse system is operating, moving alternatively between two limit switches. If a PLC power failure occurs during the traversing the machine will stop. Ideally, once the PLC regains its power the system should continue from where it left off, i.e. if the movement direction was to the left before the power down, it should continue to the left after the restoration of the power.

Explanation



The status of the latched devices (in this example M coils M600 and M601) is retained during the power down. Once the power is restored the battery backed M coils latch themselves in again, i.e. the load M600 is used to drive M600

10.6 Indexing Through Multiple Display Data Values

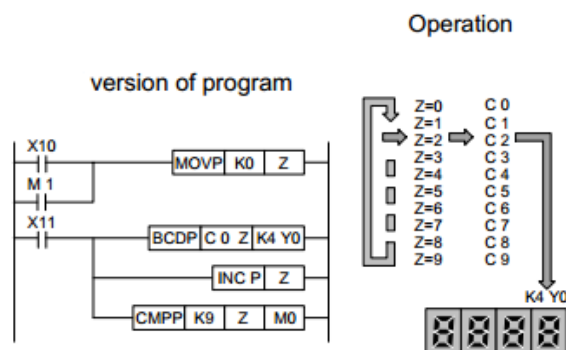
Many users unwarily fall in to the trap of only using a single seven segment display to display only a single data value. This very simple combination of applied instructions shows how a user can 'page' through multiple data values displaying each in turn.

Explanation:

The contents of 10 counters are displayed in a sequential, 'paged' operation.

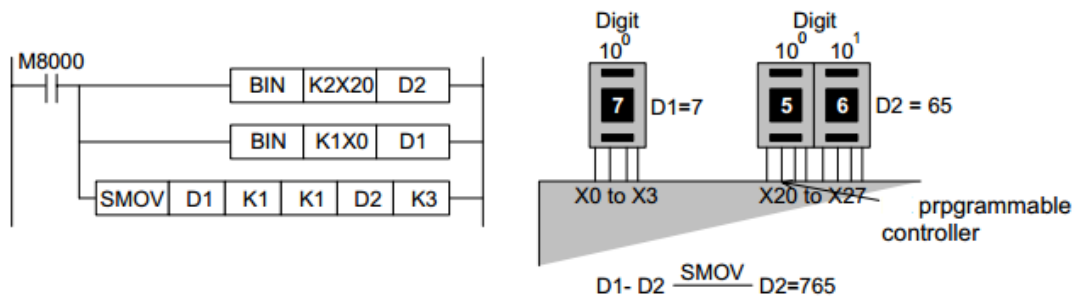
The paging action occurs every time the input X11 is received. What actually happens is that the index register Z is continually incremented until it equals 9. When this happens the comparison instruction drives M1 ON which in turn resets the current value of Z to 0 (zero). Hence, a loop effect is created with Z varying between fixed values of 0 and 9 (10 values). The Z value is used to select the next counter to be displayed on the seven segment display.

This is because the Z index modifier is used to offset the counter being read by the BCD output instruction.



10.7 Reading And Manipulating Thumbwheel Data

Data can be easily read into a programmable controller through the use of the BIN instruction. When data is read from multiple sources the data is often stored at different locations. It may be required that certain data values are combined or mixed to produce a new value. Alternatively, a certain data digit may need to be parsed from a larger data word. This kind of data handling and manipulation can be carried out by using the SMOV instruction. The example below shows how two data values (a single digit and a double digit number) are combined to make a final data value.



Explanation:

The two BIN instructions each read in one of the data values. The first value, the single digit stored in D1, is combined with the second data value D2 (currently containing 2 digits). This is performed by the SMOV instruction. The result is that the contents of D1 is written to the third digit of the contents of D2. The result is then stored back into register D2.

10.8 Measuring a High Speed Pulse Input

10.8.1 A 1 msec timer pulse measurement

Some times due to system requirements or even as a result of maintenance activities it is necessary to 'find out' how long certain input pulses are lasting for. The following program utilizes two interrupt routines to capture a pulse width and measure it with a 1 msec timer. The timer used in the example is one of the timers. However, T63 on the HCA2P/HCA2C would be used for a similar situation on that PLC.

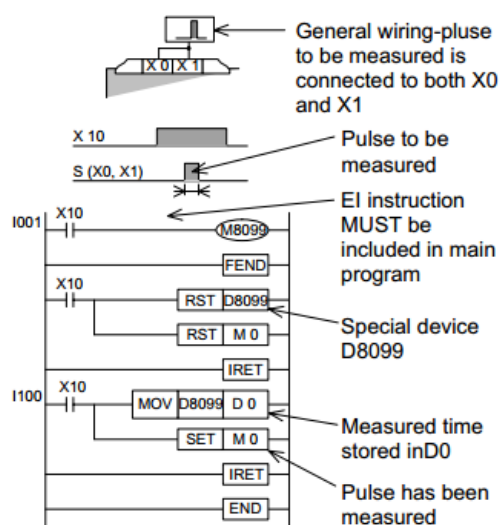
Explanation:

The 1 msec timer T246 is driven when interrupt I001 is activated. When the input to X1 is removed the current value of the timer T246 is moved to data register D0 by interrupt program I100. The operation complete flag M0 is then set ON.

Note: X10 acts as an enable/disable flag.

10.8.2 A 0.1 msec timer pulse measurement

This is a very accurate measuring process for pulse inputs. The use of a standard timer is not accurate enough in this case as the highest resolution is 1msec. Therefore, this example shows how the special high accuracy devices



M8099 and D8099 are used to capture the 0.1 msec resolution pulse data.

Explanation:

The incoming pulse is captured between two interrupt routines. These routines operate independently of each other, one on the rising edge of the pulse input and one on the falling edge of the same input. During the pulse input the contents of special register D8099 are continually moved into data register D0. Once the pulse has completed the contents of D0 can be viewed at leisure.

Please note for this high speed/accuracy mode to be active for D8099, the corresponding special auxiliary bit device M8099 must be driven ON in the main program.

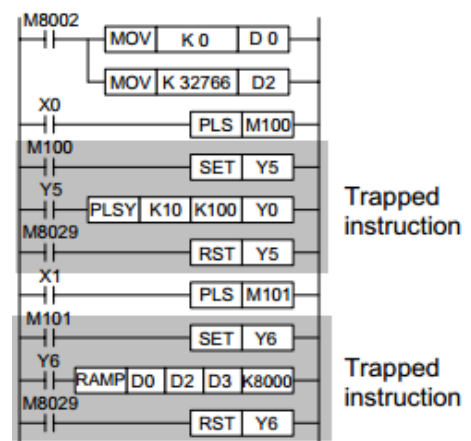
10.9 Using The Execution Complete Flag, M8029

Some of the applied instructions take more than one program scan to complete their operation.

This makes identification of the current operating state

difficult. As an aid to the programmer, certain applied instructions identify their completion by setting an operation complete flag, M8029.

Because this flag can be used by several different instructions at the same time, a method similar to the following should be used to trap the M8029 status at each of the



instructions using it:

Explanation:

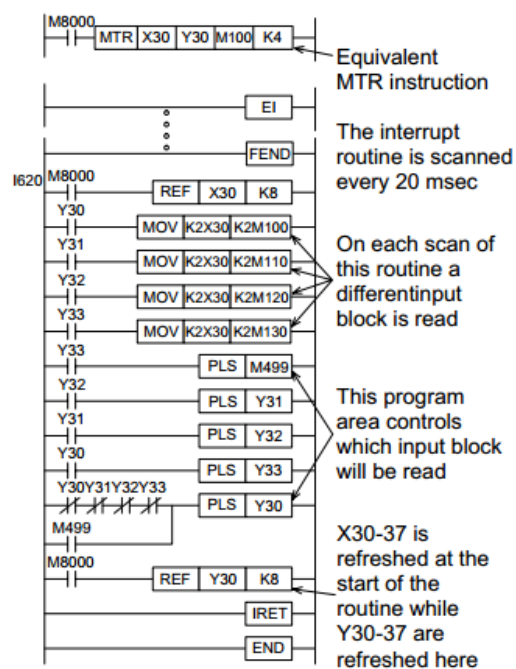
The M8029 'trapping' sequence takes advantage of the batch refresh of the PLC's.

As the program scan passes each instruction using M8029 the status of M8029 changes to reflect the current status of the instruction. Hence, by immediately resetting (or setting) the drive flag for the instruction the current operational status of the instruction is

trapped. So when the batch refresh takes place only the completed instructions are reset. The example above uses a pulse to set the drive flags so that it is easy to monitor and see when each instruction finishes (if the instructions are continuously driven it will be difficult to see when they finish!).

10.10 Creating a User Defined MTR Instruction

For users who want to have the benefits of the MTR instruction for HCFA users who want to specify more than one MTR area, this user defined MTR function will be very useful.



Explanation:

The main control of this program rests in the timer interrupt I620. This interrupt triggers every 20msec regardless of what the main program is doing. On each interruption one bank of the user defined matrix is read. The program simply consists of

reading the inputs triggered by each of the multiplexed outputs.

Thereaddataisthenstoredinsequentialsetsof auxiliary registers. Each MOV instruction reads a new bank of multiplexed inputs.

The equivalent MTR instruction is shown immediately before the 'user defined' MTR.

10.11 An Example System

Application Using STL And IST Program Control

The following illustration shows a simple 'pick and place' system utilizing a small robotic arm. The zero point has been de-fined as the uppermost and left most position accessible by the robot arm.

A normal sequence of events

A product is carried from point 'A' to point 'B' by the

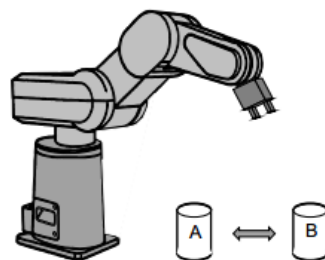
robot arm. To achieve this operation the

following sequence of events takes place:

Initial position: the robot arm is at its zero point.

1) The Robots grip is lowered to it lowest limit

- output Y0: ON, input X1: ON, output Y0: OFF.



2) The grip clamped around the product at point A

- output Y1: ON.

3) The grip, now holding the product, is raised to its upper limit

- output Y2: ON, input X2: ON, output Y2: OFF.

4) The robot arm traverses to its right most position

- output Y3: ON, input X3: ON, output Y3: OFF.

5) The grip and product are lowered to the bottom limit

- output Y0: ON, input X1: ON, output Y0: OFF.

6) The grip is unclamped and the product is released at point B

- output Y1: OFF.

7) The grip is retrieved back to its upper limit

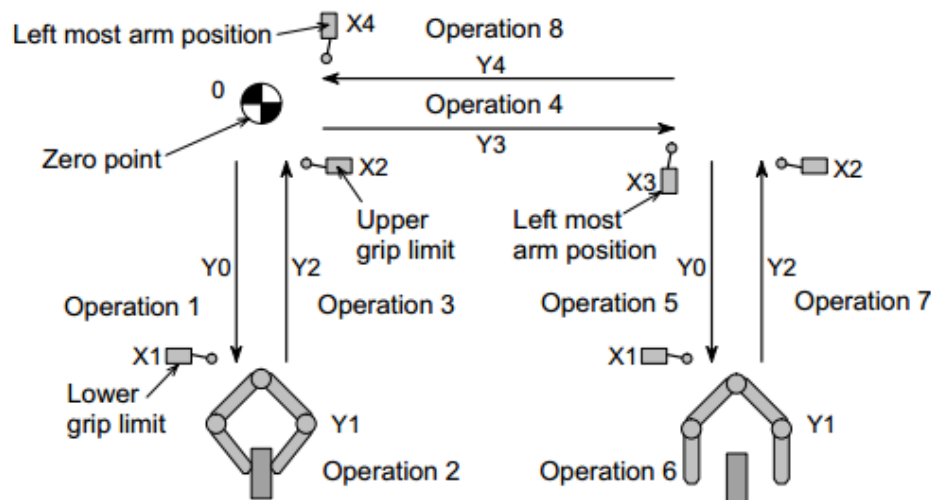
- output Y0: ON, input X2: ON, output Y0: OFF.

8) The arm traverses back to its zero point by moving to the left most limit

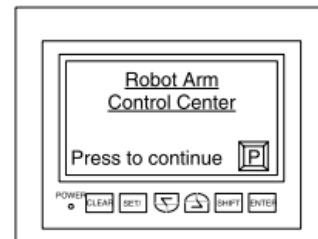
- output Y4: ON, input X4: ON, output Y4: OFF.

The cycle can then start again.

System parameters

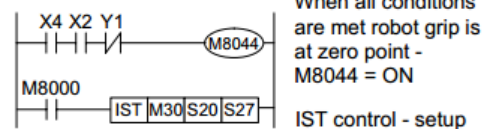


- 1) Double solenoid valves are used to control the up (Y2)/down (Y0) and right (Y3)/left (Y4) motion.
- 2) A single solenoid valve is used for the clamp (Y1)/unclamp operation.



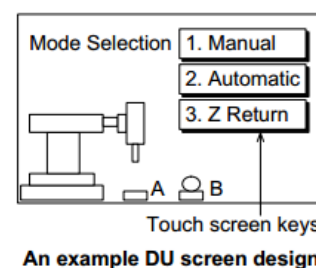
This example uses the IST instruction (FNC 60)

to control the operation mode of the robot



arm. The program shown opposite identifies how the IST instruction is written into the main program.

When the IST instruction is used there are 5 selectable modes



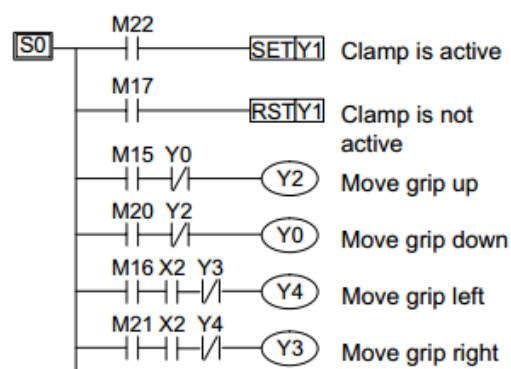
which access three separate programs. This example has the following programs associated with its modes. Each mode is selected. The screen shown opposite is the initial mode menu. Each of the menu options causes a screen jump to the selected mode. Menu options 1 and 3 also set ON auxiliary devices M30 and M31 respectively.

The active bits then trigger a screen change to the selected mode. Please note 'Automatic' has three further modes which are selected from a following screen/display.

Manual Mode:

In this mode ALL operations of the robot arm are controlled by the operator. An operation or movement is selected by pressing the corresponding option on the DUs screen (see

below). These options then trigger DU SWITCH objects which drive associated auxiliary relays within the programmable controller. The SWITCH objects should be



set to momentary so that they
 only operate when the key is pressed.

The status of the clamping action could be identified by two INDICATOR (SCR) functions on the DU unit. They could be monitoring the ON and OFF status of the clamp output Y1. Hence, when the clamp was ON a single black box opposite the ON button could appear. When the clamp is OFF the box would appear in front of the OFF button. At any one time only one box would be active.

Key assignment for DU screen opposite:

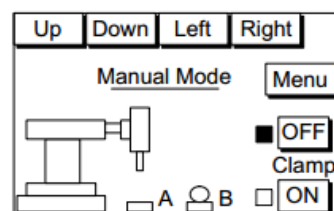
Up = M15 Down = M20

Left = M16 Right = M21

Clamp ON = M22

Clamp OFF = M17

Menu = reset M30

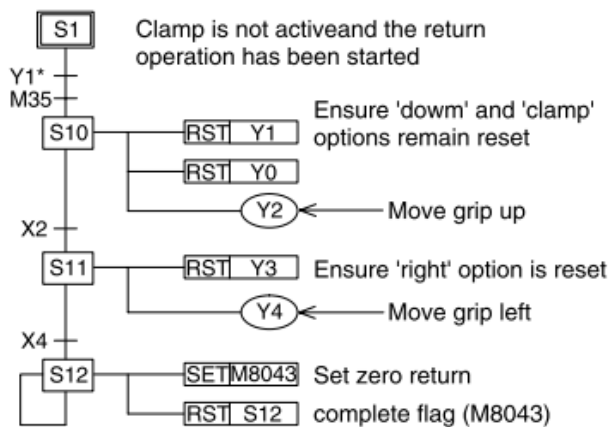


Once manual operation is completed the operator can return to the main mode selection screen by touching the 'Menu' key. This causes the manual mode bit flag, M30, to be reset. Once M30 is reset the DU screen then changes back to the desired mode selection screen.

Zero Return Mode

This mode fulfills an initialization function by returning the robot arm to a known position.

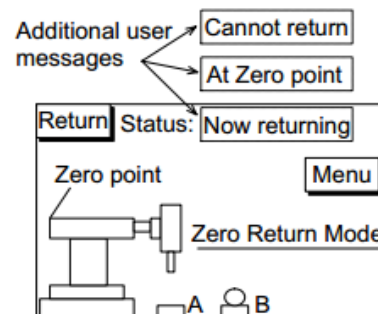
Once 'Z Return' has been selected from the mode selection screen the



bit device M35 is ON. At this point the DU screen changes to the 'zero return' screen.

The actual zero return operation will then start when the 'Return' push button is pressed (activating M25) and the robots grip is not active, i.e. Y1 is OFF (on the STL flow diagram opposite Y1 OFF is shown as Y1*).

The DU unit could be used to report back the status of the current returning operation. The example screen shown opposite uses 3 variable messages to indicate this status. The messages could be text strings stored in the PLC which are read and displayed by the DUs ASCII option.



Once the zero point has been returned to, the operator would also return to the mode selection screen. This is achieved by pressing the 'Menu' touch key. This then resets the zero return bit device M31 which allows the DU screen change to take place.

Key assignment for DU screen above:

Return = M25

Menu = reset M31

Automatic Mode

Under this option there are three further mode selections. The available modes are:

Step Mode:

- The automatic program is stepped through - operation by operation, on command by the user pressing the 'Start' button.

Cycle Mode:

- The automatic program is processed for one complete operational cycle. Each cycle is initiated by pressing the 'Start' button. If the 'Stop' button is pressed, the program is stopped immediately. To resume the cycle, the 'Start' button is pressed again.

Automatic Mode:

- A fully automatic, continuously cycling mode. The modes operation can be stopped by pressing the 'stop' button. However, this will only take effect after completion of the current cycle.

In this example these three modes are selected

by an external rotary switch. The rotary switch is

not connected to the PLC but to the I/O bus on

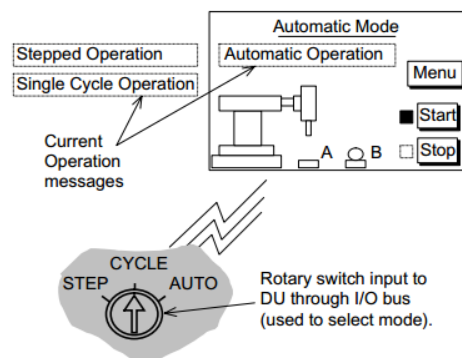
the rear of the DU unit. The use of the rotary

switch means that the selected modes are

mutually exclusive in their operation. For an operator friendly environment the

currently selected mode is displayed on the DU screen (again this could be by use of

the DUs ASCII function). The start/ stop controls are touch keys on the DU screen.



When a mode is selected the input received at the DU unit momentarily activates one of the following auxiliary relays:

Rotary switch:

position 1 'Step' - Step operation: DU input I0, controls bit device M32 position 2

'Cycle' -Single cycle operation:

DU input I1, controls bit device M33 position 3

'Auto' - Automatic operation: DU input I2, controls bit device M34

Key assignment for DU screen above:

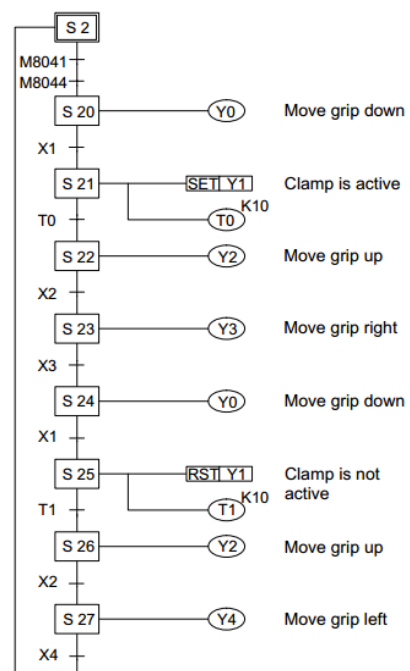
Start = M36

Stop = M37

The program run in all three mode choices is shown opposite. As noted earlier, the 'Step' mode will require an operator to press the 'Start' key to start each new STL block. This could be viewed as an additional transfer condition between each state. However, the user is not required to program this as the IST instruction controls this operation

automatically.

The 'Cycle' mode will process the program from STL step S2, all the way through until STLstepS2isencounteredagain.Once more the IST instruction ensures that only one cycle



is completed for each initial activation of the 'Start' input.

Finally as suggested by the name, 'Auto' mode will continuously cycle through the program until the 'Stop' button is pressed.

The actual halting of the program cycling will occur when the currently active cycle is completed.

Points of interest:

- a) Users of the IST instruction will be aware that only one of the operation modes should be active at one time. In this example program the isolation of 'Manual' and 'Zero return' modes by the use of separate DU control screens, and the use of a rotary switch to isolate the three automatic modes achieves this objective. Alternatively all of the operation modes could be selected by a rotary switch.
- b) For users who would like to test this example using simulator switches (i.e., without using a data access unit) the appropriate program changes are noted next to the full program listing later in this section. Alternatively, the original program could be used

with all of the input conditions being given by forcing ON the contacts with a programming device e.g. a hand held programmer, Medoc etc.

c) Special flags used in this program are:

- M8040: State transfer inhibit

- Manual mode: Always ON.

Zero return and Cycle modes: Once the 'Stop' input is given the current state is retained until the 'Start' input is received.

Step mode: This flag is OFF when the 'Start' input is ON. At all other times M8040 is ON, this enables the single STL step operation to be achieved.

Auto mode: M8040 is ON initially when the PLC is switched into RUN. It is reset when the 'Start' input is given.

- M8041: State transfer start

- Manual and Zero return modes: This flag is not used.

Step and Cycle modes: This flag is only active while the 'Start' input is received.

Auto mode: The flag is set ON after the 'Start' input is received. It is reset after the 'Stop' input is received.

- M8042: Start pulse

- This is momentarily active after the 'Start' input is received.

- M8043: Zero return complete

- This is a user activated device which should be controlled within the users program.

- M8044: At Zero position/ condition

- This is a user activated device which should be controlled within the users program.

Full program listing:

0	LD	X	4		35	STL	S	1		72	STL	S	21
1	AND	X	2		36	LD	M	35		73	SET	Y	1
2	ANI	Y	1		37	RST	M	8043		74	OUT	T	0
3	OUT	M	8044		39	ANI	Y	1				K	10
5	LD	M	8000		40	SET	S	10		77	LD	T	0
6	IST		60		42	STL	S	10		78	SET	S	22
		M	30		43	RST	Y	1		80	STL	S	22
		S	20		44	RST	Y	0		81	OUT	Y	2
		S	27		45	OUT	Y	2		82	LD	X	2
13	STL	S	0		46	LD	X	2		83	SET	S	23
14	LD	M	8044		47	SET	S	11		85	STL	S	23
15	OUT	M	8043		49	STL	S	11		86	OUT	Y	3
17	LD	M	22		50	RST	Y	3		87	LD	X	3
18	SET	Y	1		51	OUT	Y	4		88	SET	S	24
19	LD	M	17		52	LD	X	4		90	STL	S	24
20	RST	Y	1		53	SET	S	12		91	OUT	Y	0
21	LD	M	15		55	STL	S	12		92	LD	X	1
22	ANI	Y	0		56	SET	M	8043		93	SET	S	25
23	OUT	Y	2		58	RST	S	12		95	STL	S	25
24	LD	M	20			(RET)*				96	RST	Y	1
25	ANI	Y	2		60	STL	S	2		97	OUT	T	1
26	OUT	Y	0		61	LD	M	8041				K	10
27	LD	M	16		62	RST	M	8043		100	LD	T	1
28	AND	X	2		64	AND	M	8044		101	SET	S	26
29	ANI	Y	3		65	SET	S	20		103	STL	S	26
30	OUT	Y	4		67	STL	S	20		104	OUT	Y	2
31	LD	M	21		68	OUT	Y	0		105	LD	X	2
32	AND	X	2		69	LD	X	1		106	SET	S	27
33	ANI	Y	4		70	SET	S	21		108	STL	S	27
34	OUT	Y	3							109	OUT	Y	4
	(RET)*									110	LD	X	4
										111	OUT	S	2
										113	RET		
										114	END		

*: Instructions in () are not necessary

This instruction returns the program flow to STL step S2. →

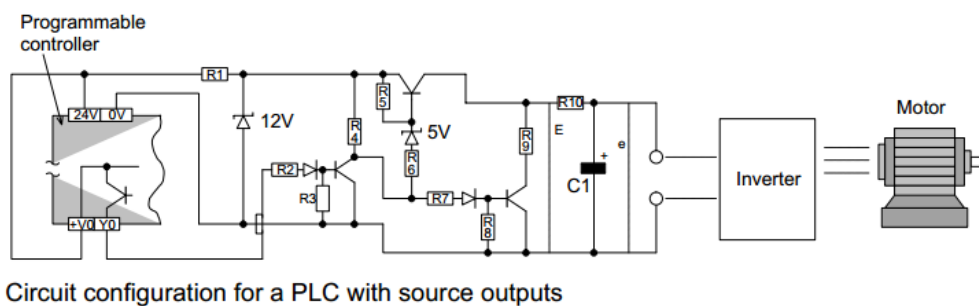
Program options:

6	IST		60		17	LD	X	12		27	LD	X	6
		X	20		19	LD	X	7		31	LD	X	11
		S	20		21	LD	X	5		36	LD	X	25
		S	27		24	LD	X	10					

10.12 Using The PWM Instruction For Motor Control

The PWM instruction may be used directly with an inverter to drive a motor. If this configuration is used the following ripple circuit will be required between the PLC's

PWM output and the inverters input terminals.



Key to component values:

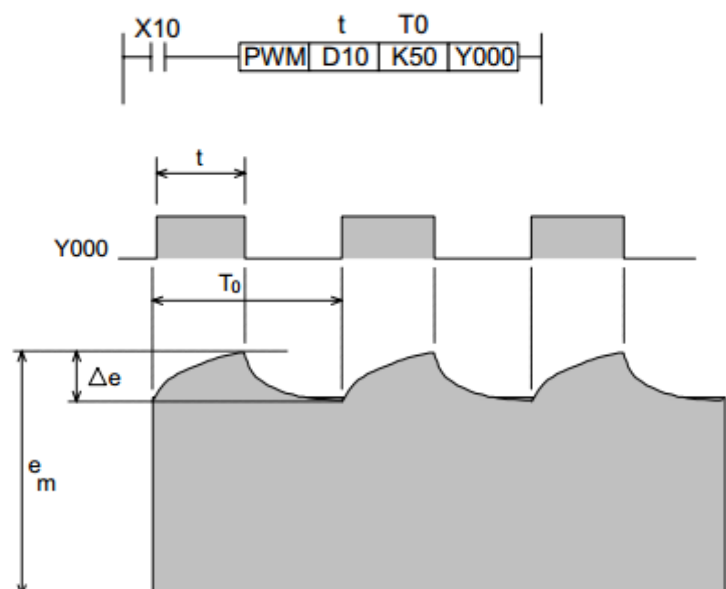
R1 - 510Ω (1/2 W) R2 - $3.3k\Omega$ (1/2 W)

R3 to R8 - $1\text{k}\Omega$ (1/4 W) R9 - 22Ω (1/4 W)

R10 - variable dependent on configuration. In this example $1\text{k}\Omega$ (1 W)

C1 - 470 μ F

Note: the values of R10 and C1 are dependent on the system configuration



Establishing system parameters and values

It is assumed that the input impedance of the inverter is of a high order. Having

established this, the values of C1 and R10 are calculated to give τ time result (in msec)

approximately 10 times bigger than the value used for T0 in the PWM instruction:

$$\tau = R10(k\Omega) \times C1(\mu F)$$

During this calculation the value of R10 must be vastly greater than the value of R9. In

the example, R9 is equal to 22 Ω , where as R10 is equal to 1k Ω . This proportion is

approximately 1:50 in favor of R10.

The maximum output voltage (to the inverter) including ripple voltage, can be found

by using the following equation:

$$e_m \approx E \frac{t}{T_0}$$

Where:

e_m = Maximum output voltage

E = pulse (square wave) output voltage (see circuit on the previous page)

t = PWM pulse duration (see previous page for reference)

T_0 = PWM cycle time for pulse (see previous page for reference)

The average output voltage (to the inverter) including ripple voltage, can be found by

using the following equation:

$$\frac{\Delta e}{e} \approx \frac{T_0 - t}{\tau} \leq \frac{T_0}{\tau}$$

Where:

Δe = the voltage value of the ripple

e = ripple output voltage

T_0 = PWM cycle time for pulse

t = PWM pulse duration

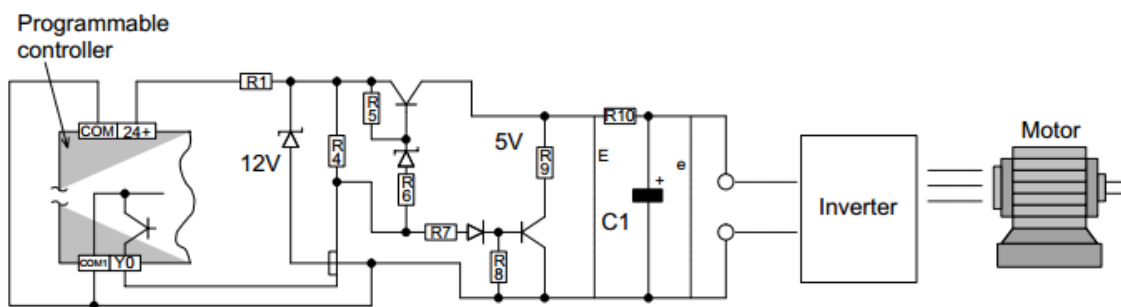
 τ = ripple circuit delay

See previous page for references.

Operation

Once the system configuration has been selected and the ripple circuit has been built to suit, the motor speed may be varied by adjusting the value of 't' in the PWM instruction.

The larger the value of 't' the faster the motor speed will rotate. However, this should be balanced with the knowledge that the faster the output signal changes the greater the ripple voltage will be. On the other hand a slowly changing output signal will have a more controlled, yet smaller ripple effect. The speed of the signal change is determined by the size of C1. A large capacitive value for C1 would give a smaller ripple effect as charge is stored and released over a longer time period.

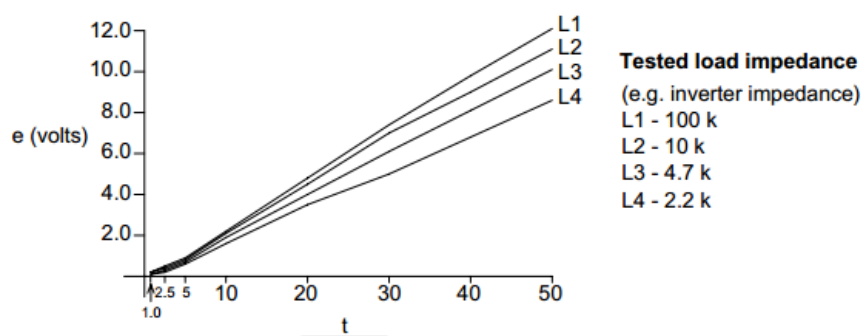


Circuit configuration for a PLC with sink outputs.

The component values are the same as stated previously

The following characteristics were noticed when the identified circuit was tested

The PWM instruction had T0 set to K50. The value for t was varied and also the load impedance was varied to provide the following characteristics graph (see over page).



The duration of the T0, time base also affects the ripple voltage. This can be clearly seen in the next set of test data:

PWM parameter setting			Measured ripple voltage
t	T0	t / T0	
100	200	0.5	1.27V
50	100		668mV
25	50		350mV
10	20		154mV
5	10		82mV

The behavior of the Sink switched circuit detailed above will be similar to that of the Source switched circuit detailed earlier.

10.13 Communication Format

10.13.1 Specification of the communication parameters:

Items such as baud rates, stop bits and parities must be identically set between the two communicating devices. The communication parameters are selected by a bit pattern which is stored in data register D8120.

D8120			
	Description	Bit (bn)status	
		0 (OFF)	1 (ON)
b0	Data length	7 bits	8 bits
b1 b2	Parity (b2, b1)	(00): No parity (01): Odd parity (11): Even parity	
b3	Stop bits	1 bit	2bits
b4 b5 b6 b7	Baud rate - bps	(b7, b6, b5, b4) (0011): 300 bps (0100): 600 bps (0101): 1200 bps (0110): 2400 bps	(b7, b6, b5, b4) (0111): 4800 bps (1000): 9600 bps (1001): 19200 bps
b8	Header character	None	D8124, Default: STX (02H)
b9	Terminator character	None	D8125, Default: ETX (03H)
b10 b11 b12	Communication Control (see timing diagrams page 10-20 onwards)	No Protocol (b12, b11, b10) (0, 0, 0): RS Instruction is not being used (RS232C interface) (0, 0, 1): Terminal mode -RS232C interface (0, 1, 0): Interlink mode - RS232C interface (0, 1, 1): Normal mode 1- RS232C, RS485(422) interfaces (RS485 HCA5 (C) only) (1, 0, 1): Normal Mode 2 - RS232C interface	
		Computer Link (b12, b11, b10) (0, 0, 0): RS485(422) interface (0, 1, 0): RS232C interface	



General note regarding the use of Data register D8120: This data register is a general set-up register for all ADP type communications. Bits 13 to 15 in the 232ADP units should not be used. When using the network with 485ADP units bits 13 to 15 should be used instead of bits 8 to 12.

10.13.2 Header and Terminator Characters

The header and terminator characters can be changed by the user to suit their requirements.

The default setting for the header stored in D8124 is 'STX' (or 02H) and the terminator default setting stored in D8125 is 'ETX' (or 03H). The header and terminator characters are automatically added to the 'send' message at the time of transmission. During a

receive cycle, data will be ignored until the header is received. Data will be continually read until either the termination character is received or the receive buffer is filled. If the buffer is filled before the termination character is received then the message is considered incomplete.

If no termination character is used, then reading will continue until the receive data buffer is full. Only at this point will a message have been accepted and complete.

There is no further buffering of any communications, hence if more data is sent than the available destination buffer size then the excess will be lost once the buffer is full.

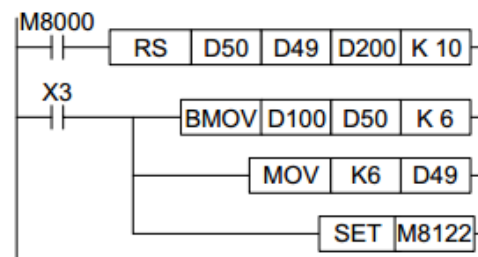
It is therefore very important to specify the receive buffer length the same size as the longest message to be received.

Events to complete a transmission:

The RS instruction should be set up and active.

The data to be transmitted should be

moved into the transmission data buffer. If



a variable is being used to identify the message length in the RS instruction this should be set to the new message length. The send flag M8122 should then be SET ON.

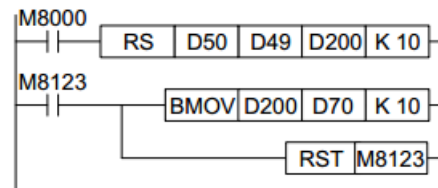
This will automatically reset once the message has been sent. Please see the example program right.

Events encountered when receiving a message:

The RS instruction should be set up and active. Once data is being received and an attempt is made to send out data, the special M

flag M8121 is set ON to indicate the

transmission will be delayed. Once the



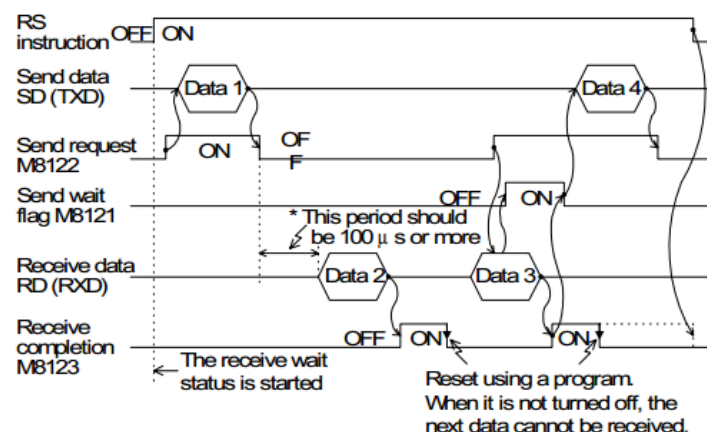
'incoming' message is completely received the message received flag M8123 is set ON.

At the same time if M8121 was ON it is automatically reset allowing further messages (delayed or otherwise) to be transmitted.

It is advisable to move the received data out of the received data buffer as soon as possible. Once this is complete M8123 should be reset by the user. This is then ready to send a message or to await receipt of a new message.

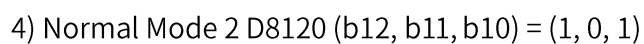
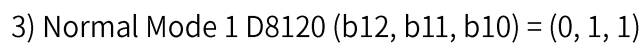
10.13.3 Timing diagrams for communications:

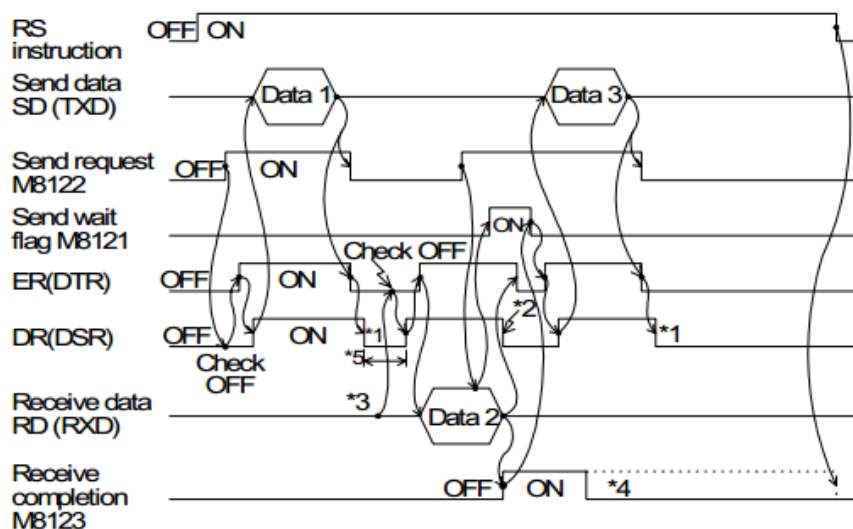
1) No Handshaking D8120 (b12, b11, b10) = (0, 0, 0)



2) Terminal mode D8120 (b12, b11, b10) = (0, 0, 1)

a) Send Only





10.13.4 8 bit or 16 bit communications.

This is toggled using the Auxiliary relay M8161. When this relay is OFF 16 bit communications takes place. This actually means that both bytes of a 16 bit data device are used in both the transmission and the receipt of messages. If the M8161 device is activated then 8 bit mode is selected. In this mode only the lower 8 bits (or byte) is used to perform the transmissionreceiving actions. The toggling of the M8161 device should only occur when the RS instruction is not active, i.e. it is OFF.

When a buffer area is specified in the RS instruction it is important to check whether 8 or 16bit mode has been selected, i.e. a buffer area specified as D50 K3 would produce the following results.....

16 bit mode - M8161 = OFF		
Data register	High byte	Low byte
D50	X	F
D51		0

8 bit mode - M8161 = ON		
Data register	High byte	Low byte
D50		F
D51		X
D52		0

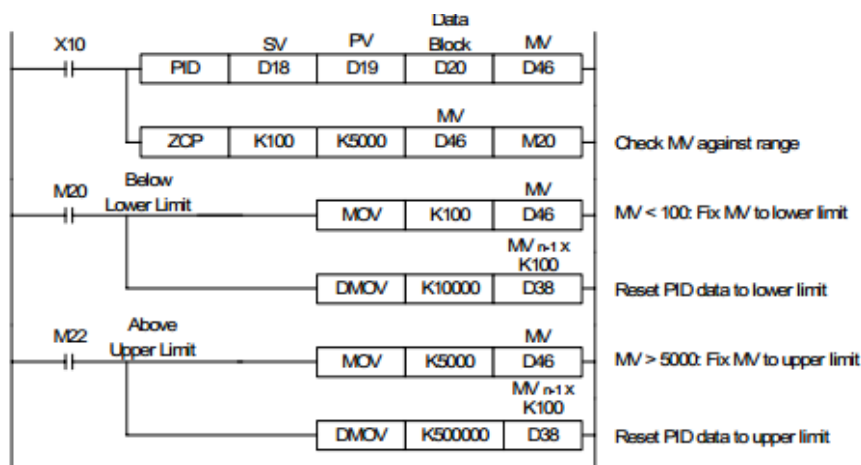
② General note regarding hardware: Information regarding pin outs of the respective ADP special function blocks can be found along with wiring details in the appropriate hardware manuals.

10.14 PID Programming Techniques

10.14.1 Keeping MV within a set range

In the reserved registers of the PID data block S 3+18 and S3+19 form a double word device that contains the previous MV x K100. The following program uses this to keep MV under control when it exceeds the operating limits.

Example Program to keep MV in the range K100 to K5000



If data registers are used to hold the limit values, it is possible to use a MUL instruction instead of the DMOV. E.g. When D50 is upper limit use: MUL D50 K100 D38 because the result of MUL is already a double word DMUL is not needed.

Resetting (S3+19, S3+18) in this way prevents runaway, which occurs if only MV is changed.

10.14.2 Manual/Automatic change over

In order to switch from automatic (PID) control to manual control and back to automatic it is necessary for the PID process to perform 'Manual Tracking'. Although the HC PID instruction does not have a manual tracking feature there are two methods that can be used to make the switch from manual back to automatic as trouble free as possible.

To understand the reason for the two methods the following should be noted. The PID instruction sets its initial output value based on the initial value of the output register. When the PID instruction is switched on it can only do P as it has only 1 data reading. On the first reading the current value of the output register is used as ΔMV . Thereafter the previous output value is used (stored in S3+18, S3 +19).

After the next reading PI can be calculated and from the third reading full PID is performed.

Please see section 5.98, PID (FNC 88), for the complete equations.

Method

It is recommended that if manual to auto switching is desired that the PID instruction is switched off during manual operation and the operator controls the value of the MV register (the Output Value). When returning to auto mode, the PID instruction is switched on again and uses the last MV input by the operator during the first PID calculation. After 3 readings full PID will be operating and the process should be under

control quickly. (Assuming that manual control did not cause a move too far from the Set Point.

10.14.3 Using the PID alarm signals

Included as part of the data block there are four alarm values. These set the maximum positive and negative change that should occur to MV and PV. The PID alarm signals are used to warn of the system going out of control.

When the system is starting from cold it is usually not good to include the Derivative numbers of the in the calculation; the changes to PV are large and the Derivative introduces too much correction. Also, if the system starts to move rapidly away from the SV then sometimes the use of D can over correct and cause chasing.

By having an 'alarm' flag for the change in PV and MV it is possible to monitor the state of the system and adjust the PID parameters to appropriate settings.

When the system is close to the SP the changes in PV (and MV) should be minimal. In this situation using full PID is very useful in keeping the system close to the SP. (Full PID is appropriate).

However, if the conditions change (e.g. opening a refrigerator door, adding ingredients to a mixture, cold start, etc.) the system reacts. In some cases (especially cold start) the reaction is too much for the D to be useful (PI or sometimes just P only is better). In these cases the alarm flags can be used to change to PI control until the system returns to a more stable condition, when full PID can then be used.

Basically, rather than use actual values of the PV to determine the change over point from PI to PID (or PID to PI), use the size of the change in PV (or MV). This means changes to the Set Point do not require different ranges for the PI - PID change over point (at least, in theory).

10.14.4 Other tips for PID programming

- It is recommended that an input value for PV is read before the PID is activated. Otherwise, the PID will see a big change from 0 to the first value and calculate as if a big error is occurring.
- The PID instruction is not interrupt processed. It is scan dependent and as such the sampling can not occur faster than the HC scan time. It is recommended that TS is set to a multiple of the program scan time.
- To keep timing errors to a minimum it is recommended that constant scan is used.
- To improve sampling rates it is possible to put the PID instruction inside a timer interrupt routine.
- It is better to have the PID only perform P until the input value (PV) reaches the working range.
- When setting up it is a good idea to monitor the input and output of the PID instruction and check that they are about the expected values.
- If the PID system is not operating properly check the error flags for PID errors (D8067).

10.15 Pre-tuning operation

10.15.1 Variable Constants

The Pre-tuning operation can be used to automatically set values for the following variables:

- The direction of the process; Forward or Reverse (S 3+1, bit 0)
- The proportional gain constant; KP(S3+3)
- The integral time constant; TI(S3+4)
- The derivative time constant; TD(S3+6)

Setting bit 4 of S3+1 starts the pre-tuning process. Before starting, set all values that are not set by the pre-tuning operation: the sample time, Ts (S3 +0); the input filter α (S 3+2); the Derivative gain, KD(S 3+5); the Set Point, SV (S1); and any alarm or limit values, (S 3 +20-23).

The Pre-tuning operation measures how fast the system will correct itself when in error. Because the P, I, and D equations all react with differing speed, the initial error must be large so that effective calculations can be made for each type of equation.

The difference in values between SP and PV_{nf} must be a minimum of 150 for the Pre-tuning to operate effectively. If this is not the case, then please change SV to a suitable value for the purpose of pre-tuning.

The system keeps the output value (MV) at the initial value, monitoring the process value until it reaches one third of the way to the Set Point. At this point the pre-tuning

flag (bit 4) is reset and normal PID operation resumes. SV can be returned to the normal setting without turning the PID command Off.

During the course of normal operation, the Pre-tuning will NOT automatically set new values if the SV is changed. The PID command must be turned Off, and the Pre-Tuning function restarted if it is necessary to use the Pre-tune function to calculate new values.

- **Caution:** The Pre-tuning can be used as many times as necessary. Because the flag resets, the set bit can be turned On again and new values will be calculated. If the system is running an oven heater and the SV is reduced from 250 to 200 C, the temperature must drop below 200 or the “Forward/Reverse” flag will be set in the wrong direction. In addition, the system error value must be large for the pre-tune variable calculations to work correctly.



- **Note:** Set the sampling time to greater than 1 second (1000 ms) during the pre-tuning operation. It is recommended that the sampling time is generally set to a value much greater than the program scan time.

- **Note:** The system should be in a stable condition before starting the pre-tuning operation.

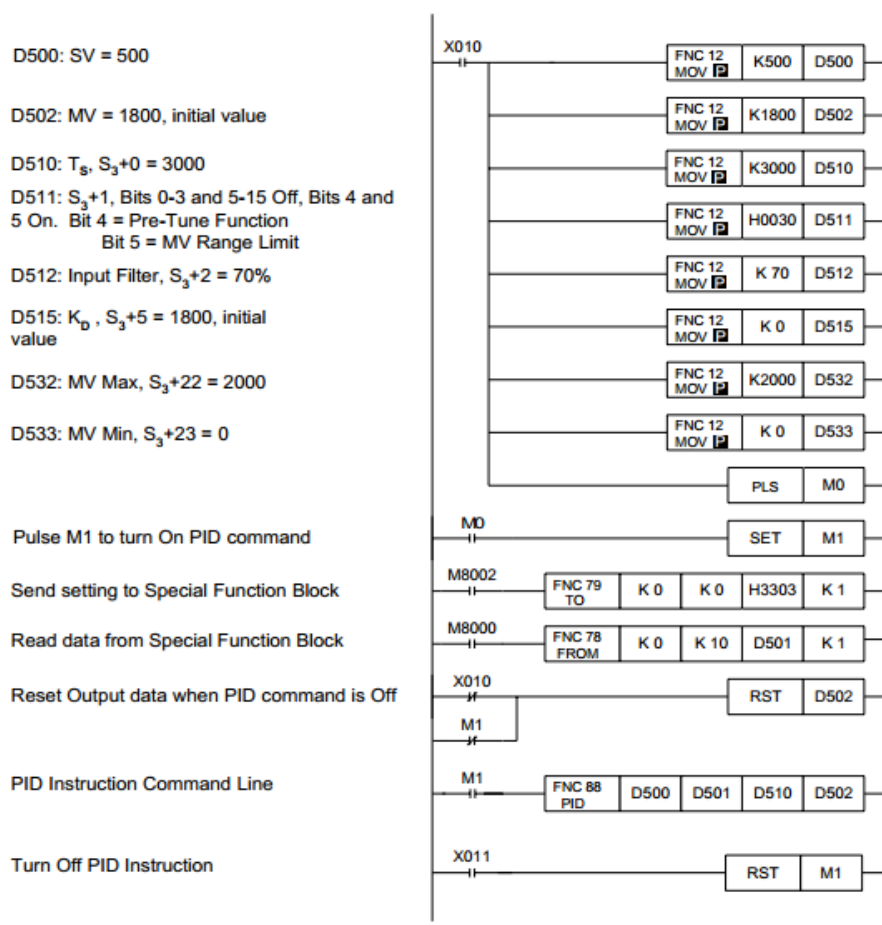
An unstable system can cause the Pre-tuning operation to produce invalid results. (e.g. opening a refrigerator door, adding ingredients to a mixture, cold start, etc.)

- **Note:** Even though Pre-tuning can set the above mentioned variables, additional logic may be needed in the program to "scale" all operating values to those capable of

being processed by the special function devices being used.

10.16 Example Autotuning Program

The following programming code is an example of how to set up the Pre-Tuning function.





— 禾川股份 — HCFA CORPORATION



— 禾川股份 —