6 Applied Instructions

M, VB and VH Series PLC has many applied instructions, each instruction has its specific function. PLC will achieve a complicated control system and diminish programming codes and programming development time effectively by using of these instructions. We hope readers will have an in-depth understanding of the applied instructions and make the best use of them.

6-1 Applied Instruction Table

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
Program Flow	00		CJ	P	Conditional jump	○	○	○	110
	01		CALL	P	Call subroutine	○	○	○	111
	02		SRET		Subroutine return	○	○	○	111
	03		IRET		Interrupt return	○	○	○	112
	04		EI		Enable interrupt	○	○	○	112
	05		DI		Disable interrupt	○	○	○	112
	06		FEND		First end	○	○	○	113
	07		WDT	P	Watch Dog Timer refresh	○	○	○	114
	08		FOR		Start of a FOR-NEXT loop	○	○	○	115
	09		NEXT		End of a FOR-NEXT loop	○	○	○	115
Compare and Move	10	D	CMP	P	Compare	○	○	○	118
	11	D	ZCP	P	Zone compare	○	○	○	119
	12	D	MOV	P	Move	○	○	○	120
	13		SMOV	P	Shift move	○	○	○	121
	14	D	CML	P	Compliment	○	○	○	122
	15		BMOV	P	Block move	○	○	○	123
	16	D	FMOV	P	Fill move	○	○	○	124
	17	D	XCH	P	Exchange	○	○	○	125
	18	D	BCD	P	Converts BIN → BCD	○	○	○	126
	19	D	BIN	P	Converts BCD → BIN	○	○	○	126
Arithmetic and Logical Operations	20	D	ADD	P	Addition (S1)+(S2) → (D)	○	○	○	128
	21	D	SUB	P	Subtraction (S1) − (S2) → (D)	○	○	○	129
	22	D	MUL	P	Multiplication (S1)×(S2) → (D+1,D)	○	○	○	130
	23	D	DIV	P	Division (S1)÷(S2) → (D), (D+1)	○	○	○	131
	24	D	INC	P	Increment (D)+1 → (D)	○	○	○	132
	25	D	DEC	P	Decrement (D)−1 → (D)	○	○	○	132
	26	D	WAND	P	Logic word AND (S1) ∧(S2) → (D)	○	○	○	133
	27	D	WOR	P	Logic word OR (S1) ∨(S2) → (D)	○	○	○	133
	28	D	WXOR	P	Logic word exclusive OR (S1)⩛(S2) → (D)	○	○	○	133
	29	D	NEG	P	Negation ($\overline{D}$)+1 → (D)	○	○		134
Rotary and Shift	30	D	ROR	P	Rotation Right	○	○	○	136
	31	D	ROL	P	Rotation Left	○	○	○	136
	32	D	RCR	P	Rotation Right with carry	○	○	○	137
	33	D	RCL	P	Rotation Left with carry	○	○	○	137
	34		SFTR	P	Bit shift Right	○	○	○	138
	35		SFTL	P	Bit shift Left	○	○	○	138
	36		WSFR	P	Word shift Right	○	○		139
	37		WSFL	P	Word shift Left	○	○		140
	38		SFWR	P	Shift register write (FIFO Write)	○	○	○	141
	39		SFRD	P	Shift register read (FIFO Read)	○	○	○	142
Data Operation	40		ZRST	P	Zone reset	○	○	○	144
	41		DECO	P	Decode	○	○	○	145
	42		ENCO	P	Encode	○	○	○	146
	43	D	SUM	P	The sum of active bits	○	○		147
	44	D	BON	P	Check specified bit status	○	○		148
	45	D	MEAN	P	Mean	○	○		149
	46		ANS		Timed annunciator set	○	○		150
	47		ANR	P	Annunciator reset	○	○		150
	48	D	SQR	P	Square root	○	○		152
	49	D	FLT	P	BIN integer → Binary floating point format	○	○		153

＊ D ~ A32 bit mode instruction option. ＊ P ~ Pulse (signal) operation option.

＊ ○ ~ The applicable PLC type

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
High-speed Processing	50		REF	P	I/O refresh	○	○	○	156
	51		REFF	P	I/O refresh and filter adjust	○	○		157
	52		MTR		Input matrix	○	○		158
	53	D	HSCS		High Speed Counter set	○	○	○	159
	54	D	HSCR		High Speed Counter reset	○	○	○	161
	55	D	HSZ		High Speed Counter zone compare	○	○		162
	56		SPD		Speed detection	○	○	○	167
	57	D	PLSY		Pulse Y output	○	○	○	168
	58		PWM		Pulse width modulation	○	○	○	169
	59	D	PLSR		Variable speed of Pulse output		○	○	170
Handy Instruction									
	61	D	SER	P	Search	○	○		174
	62	D	ABSD		Absolute Drum sequencer	○	○	○	175
	63		INCD		Incremental Drum sequencer	○	○	○	177
	64		TTMR		Teaching Timer	○	○		178
	65		STMR		Special Timer	○	○		179
	66		ALT	P	Alternate state	○	○	○	180
	67		RAMP		Ramp variable value	○	○	○	181
	69		SORT		Sort data	○	○		183
External Setting and Display	70	D	TKY		Ten Key input	○	○		186
	71	D	HKY		Hexadecimal Key input	○	○		187
	72		DSW		Digital Switch (Thumbwheel input)	○	○		189
	73		SEGD	P	Seven Segment Decoder	○	○	○	190
	74		SEGL		Seven Segment with Latch	○	○		191
	76		ASC		ASCII code Convert	○	○		193
	77		PR		Print	○	○		194
	78	D	FROM	P	Read from a special function block	○	○		195
	79	D	TO	P	Write to a special function block	○	○		195
External Serial Comm-unications	80		RS		Serial communication instruction	○	○	○	198
	81	D	PRUN	P	Parallel Run	○	○		202
	82		ASCI	P	Converts HEX → ASCII	○	○	○	203
	83		HEX	P	Converts ASCII → HEX	○	○	○	204
	84		CCD	P	Check Code	○	○	○	205
	85		VRRD	P	VR volume read	○	○	○	206
	86		VRSC	P	VR volume scale	○	○	○	207
	88		PID		PID control loop		○		352
	89		LINK		Easy Link communication	○	○		208
	149		MBUS		MODBUS communication		○	○	370
Floating Point	110	D	ECMP	P	Compares two BIN floating point values		○		214
	111	D	EZCP	P	Compares a BIN float range with a BIN float value		○		215
	118	D	EBCD	P	Converts BIN floating point format to DEC format		○		216
	119	D	EBIN	P	Converts DEC format to BIN floating point format		○		216
	120	D	EADD	P	Adds up two BIN floating point numbers		○		217
	121	D	ESUB	P	Subtracts one BIN floating point number from another		○		218
	122	D	EMUL	P	Multiplies two BIN floating point numbers		○		219
	123	D	EDIV	P	Divides one BIN floating point number from another		○		220
	127	D	ESQR	P	Square root of a BIN floating point value		○		221
	129	D	INT	P	BIN floating point → BIN integer format		○		222
	130	D	SIN	P	Calculates the sine of a BIN floating point value		○		223
	131	D	COS	P	Calculates the cosine of a BIN floating point value		○		224
	132	D	TAN	P	Calculates the tangent of a BIN floating point value		○		225

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
Others	90		DBRD	P	Reads data from the data bank	○	○		228
	91		DBWR	P	Writes data into the data bank	○	○		229
	147	D	SWAP	P	Swaps high/low byte	○	○		230
	169	D	HOUR		Operational Hour meter		○		376
	176		TFT		Timer (10 ms)	○	○	○	231
	177		TFH		Timer (100 ms)	○	○	○	232
	178		TFK		Timer (1 sec.)	○	○	○	233
Position Control	155	D	ABS		Absolute current value read		VB1		253
	156	D	ZRN		Zero position return		VB1		254
	157	D	PLSV		Pulse variable output		VB1		255
	158	D	DRVI		Drive to increment		VB1		256
	159	D	DRVA		Drive to absolute		VB1		257
Time & Convert	160		TCMP	P	Compare two times	○	○		236
	161		TZCP	P	Compare a time to a specified time range	○	○		237
	162		TADD	P	Adds ups two time values to get a new time	○	○		238
	163		TSUB	P	Subtracts one time value from another to get a new time	○	○		239
	166		TRD	P	Reads the RTC current value to a group of registers	○	○		240
	167		TWR	P	Sets the RTC to the value stored in a group of registers	○	○	○	241
	170	D	GRY	P	Converts BIN → Gray code	○	○		242
	171	D	GBIN	P	Converts Gray code → BIN	○	○		243
In-line Comparisons	224	D	LD=		Initial comparison contact. Active when (S1)=(S2)		○	○	246
	225	D	LD>		Initial comparison contact. Active when (S1)>(S2)		○	○	246
	226	D	LD<		Initial comparison contact. Active when (S1)<(S2)		○	○	246
	228	D	LD<>		Initial comparison contact. Active when (S1)≠(S2)		○	○	246
	229	D	LD<=		Initial comparison contact. Active when (S1)≤(S2)		○	○	246
	230	D	LD>=		Initial comparison contact. Active when (S1)≥(S2)		○	○	246
	232	D	AND=		Serial comparison contact. Active when (S1)=(S2)		○	○	246
	233	D	AND>		Serial comparison contact. Active when (S1)>(S2)		○	○	246
	234	D	AND<		Serial comparison contact. Active when (S1)<(S2)		○	○	246
	236	D	AND<>		Serial comparison contact. Active when (S1)≠(S2)		○	○	246
	237	D	AND<=		Serial comparison contact. Active when (S1)≤(S2)		○	○	246
	238	D	AND>=		Serial comparison contact. Active when (S1)≥(S2)		○	○	246
	240	D	OR=		Parallel comparison contact. Active when (S1)=(S2)		○	○	246
	241	D	OR>		Parallel comparison contact. Active when (S1)>(S2)		○	○	246
	242	D	OR<		Parallel comparison contact. Active when (S1)<(S2)		○	○	246
	244	D	OR<>		Parallel comparison contact. Active when (S1)≠(S2)		○	○	246
	245	D	OR<=		Parallel comparison contact. Active when (S1)≤(S2)		○	○	246
	246	D	OR>=		Parallel comparison contact. Active when (S1)≥(S2)		○	○	246
Newly added instructions	92		TPID		Temperature PID Control		○		363

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
A	20	D	ADD	P	Addition (S1)+(S2) → (D)	○	○	○	128
	46		ANS		Timed annunciator set	○	○		150
	47		ANR	P	Annunciator reset	○	○		150
	62	D	ABSD		Absolute Drum sequencer	○	○	○	175
	66		ALT	P	Alternate state	○	○	○	180
	76		ASC		ASCII code Convert	○	○		193
	82		ASCI	P	Converts HEX → ASCII	○	○	○	203
	155	D	ABS		Absolute current value read		VB1		253
	232	D	AND=		Serial comparison contact. Active when (S1)=(S2)		○	○	246
	233	D	AND>		Serial comparison contact. Active when (S1)>(S2)		○	○	246
	234	D	AND<		Serial comparison contact. Active when (S1)<(S2)		○	○	246
	236	D	AND<>		Serial comparison contact. Active when (S1)≠(S2)		○	○	246
	237	D	AND<=		Serial comparison contact. Active when (S1)≤(S2)		○	○	246
	238	D	AND>=		Serial comparison contact. Active when (S1)≥(S2)		○	○	246
B	15		BMOV	P	Block move	○	○	○	123
	18	D	BCD	P	Converts BIN → BCD	○	○	○	126
	19	D	BIN	P	Converts BCD → BIN	○	○	○	126
	44	D	BON	P	Check specified bit status	○	○		148
C	00		CJ	P	Conditional jump	○	○	○	110
	01		CALL	P	Call subroutine	○	○	○	111
	10	D	CMP	P	Compare	○	○	○	118
	14	D	CML	P	Compliment	○	○	○	122
	84		CCD	P	Check Code	○	○	○	205
	131	D	COS	P	Calculates the cosine of a BIN floating point value		○		224
D	05		DI		Disable interrupt	○	○	○	112
	23	D	DIV	P	Division (S1)÷(S2) → (D), (D+1)	○	○	○	131
	25	D	DEC	P	Decrement (D)−1 → (D)	○	○	○	132
	41		DECO	P	Decode	○	○	○	145
	72		DSW		Digital Switch (Thumbwheel input)	○	○		189
	90		DBRD	P	Reads data from the data bank	○	○		228
	91		DBWR	P	Writes data into the data bank	○	○		229
	158	D	DRVI		Drive to increment		VB1		256
	159	D	DRVA		Drive to absolute		VB1		257
E	04		EI		Enable interrupt	○	○	○	112
	42		ENCO	P	Encode	○	○	○	146
	110	D	ECMP	P	Compares two BIN floating point values		○		214
	111	D	EZCP	P	Compares a BIN float range with a BIN float value		○		215
	118	D	EBCD	P	Converts BIN floating point format to DEC format		○		216
	119	D	EBIN	P	Converts DEC format to BIN floating point format		○		216
	120	D	EADD	P	Adds up two BIN floating point numbers		○		217
	121	D	ESUB	P	Subtracts one BIN floating point number from another		○		218
	122	D	EMUL	P	Multiplies two BIN floating point numbers		○		219
	123	D	EDIV	P	Divides one BIN floating point number from another		○		220
	127	D	ESQR	P	Square root of a BIN floating point value		○		221
F	06		FEND		First end	○	○	○	113
	08		FOR		Start of a FOR-NEXT loop	○	○	○	115
	16	D	FMOV	P	Fill move	○	○	○	124
	49	D	FLT	P	BIN integer → Binary floating point format	○	○		153
	78	D	FROM	P	Read from a special function block	○	○		195
G	170	D	GRY	P	Converts BIN → Gray code	○	○		242
	171	D	GBIN	P	Converts Gray code → BIN	○	○		243
H	53	D	HSCS		High Speed Counter set	○	○	○	159
	54	D	HSCR		High Speed Counter reset	○	○	○	161
	55	D	HSZ		High Speed Counter zone compare	○	○		162
	71	D	HKY		Hexadecimal Key input	○	○		187
	83		HEX	P	Converts ASCII → HEX	○	○	○	204
	169	D	HOUR		Operational Hour meter		○		376

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
I	03		IRET		Interrupt return	○	○	○	112
	24	D	INC	P	Increment (D)+1 → (D)	○	○	○	132
	63		INCD		Incremental Drum sequencer	○	○	○	177
	129	D	INT	P	BIN floating point → BIN integer format		○		222
L	89		LINK		Easy Link communication	○	○		208
	224	D	LD=		Initial comparison contact. Active when (S1)=(S2)		○	○	246
	225	D	LD>		Initial comparison contact. Active when (S1)>(S2)		○	○	246
	226	D	LD<		Initial comparison contact. Active when (S1)<(S2)		○	○	246
	228	D	LD<>		Initial comparison contact. Active when (S1)≠(S2)		○	○	246
	229	D	LD<=		Initial comparison contact. Active when (S1)≤(S2)		○	○	246
	230	D	LD>=		Initial comparison contact. Active when (S1)≥(S2)		○	○	246
M	12	D	MOV	P	Move	○	○	○	120
	22	D	MUL	P	Multiplication (S1)×(S2) → (D+1.D)	○	○	○	130
	45	D	MEAN	P	Mean	○	○		149
	52		MTR		Input matrix	○	○		158
	149		MBUS		MODBUS communication		○	○	370
N	09		NEXT		End of a FOR-NEXT loop	○	○	○	115
	29	D	NEG	P	Negation ($\overline{D}$)+1 → (D)	○	○		134
O	240	D	OR=		Parallel comparison contact. Active when (S1)=(S2)		○	○	246
	241	D	OR>		Parallel comparison contact. Active when (S1)>(S2)		○	○	246
	242	D	OR<		Parallel comparison contact. Active when (S1)<(S2)		○	○	246
	244	D	OR<>		Parallel comparison contact. Active when (S1)≠(S2)		○	○	246
	245	D	OR<=		Parallel comparison contact. Active when (S1)≤(S2)		○	○	246
	246	D	OR>=		Parallel comparison contact. Active when (S1)≥(S2)		○	○	246
P	57	D	PLSY		Pulse Y output	○	○	○	168
	58		PWM		Pulse width modulation	○	○	○	169
	59	D	PLSR		Variable speed of Pulse output		○	○	170
	77		PR		Print	○	○		194
	81	D	PRUN	P	Parallel Run	○	○		202
	88		PID		PID control loop		○		352
	157	D	PLSV		Pulse variable output		VB1		255
R	30	D	ROR	P	Rotation Right	○	○	○	136
	31	D	ROL	P	Rotation Left	○	○	○	136
	32	D	RCR	P	Rotation Right with carry	○	○	○	137
	33	D	RCL	P	Rotation Left with carry	○	○	○	137
	50		REF	P	I/O refresh	○	○	○	156
	51		REFF	P	I/O refresh and filter adjust	○	○		157
	67		RAMP		Ramp variable value	○	○	○	181
	80		RS		Serial communication instruction	○	○	○	198
S	02		SRET		Subroutine return	○	○	○	111
	13		SMOV	P	Shift move	○	○	○	121
	21	D	SUB	P	Subtraction (S1) – (S2) → (D)	○	○	○	129
	34		SFTR	P	Bit shift Right	○	○	○	138
	35		SFTL	P	Bit shift Left	○	○	○	138
	38		SFWR	P	Shift register write (FIFO Write)	○	○	○	141
	39		SFRD	P	Shift register read (FIFO Read)	○	○	○	142
	43	D	SUM	P	The sum of active bits	○	○		147
	48	D	SQR	P	Square root	○	○		152
	56		SPD		Speed detection	○	○	○	167
	61	D	SER	P	Search	○	○		174
	65		STMR		Special Timer	○	○		179
	69		SORT		Sort data	○	○		183
	73		SEGD	P	Seven Segment Decoder	○	○	○	190
	74		SEGL		Seven Segment with Latch	○	○		191
	130	D	SIN	P	Calculates the sine of a BIN floating point value		○		223
	147	D	SWAP	P	Swaps high/low byte	○	○		230

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
T	64		TTMR		Teaching Timer	○	○		178
	70	D	TKY		Ten Key input	○	○		186
	79	D	TO	P	Write to a special function block	○	○		195
	92		TPID		Temperature PID Control		○		363
	132	D	TAN	P	Calculates the tangent of a BIN floating point value		○		225
	160		TCMP	P	Compare two times	○	○		236
	161		TZCP	P	Compare a time to a specified time range	○	○		237
	162		TADD	P	Adds ups two time values to get a new time	○	○		238
	163		TSUB	P	Subtracts one time value from another to get a new time	○	○		239
	166		TRD	P	Reads the RTC current value to a group of registers	○	○		240
	167		TWR	P	Sets the RTC to the value stored in a group of registers	○	○	○	241
	176		TFT		Timer (10 ms)	○	○	○	231
	177		TFH		Timer (100 ms)	○	○	○	232
	178		TFK		Timer (1 sec.)	○	○	○	233
V	85		VRRD	P	VR volume read	○	○	○	206
	86		VRSC	P	VR volume scale	○	○	○	207
W	07		WDT	P	Watch Dog Timer refresh	○	○	○	114
	26	D	WAND	P	Logic word AND (S1) ∧(S2) → (D)	○	○	○	133
	27	D	WOR	P	Logic word OR (S1) ∨(S2) → (D)	○	○	○	133
	28	D	WXOR	P	Logic word exclusive OR (S1)⩛(S2) → (D)	○	○	○	133
	36		WSFR	P	Word shift Right	○	○		139
	37		WSFL	P	Word shift Left	○	○		140
X	17	D	XCH	P	Exchange	○	○	○	125
Z	11	D	ZCP	P	Zone compare	○	○	○	119
	40		ZRST	P	Zone reset	○	○	○	144
	156	D	ZRN		Zero position return		VB1		254

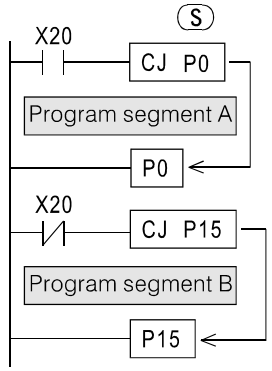
6-2 Program Flow Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
00		CJ	P	Conditional Jump	○	○	○
01		CALL	P	Call Subroutine	○	○	○
02		SRET		Subroutine Return	○	○	○
03		IRET		Interrupt Return	○	○	○
04		EI		Enable Interrupt	○	○	○
05		DI		Disable Interrupt	○	○	○
06		FEND		First End	○	○	○
07		WDT	P	Watch Dog Timer Refresh	○	○	○
08		FOR		Start of a FOR-NEXT Loop	○	○	○
09		NEXT		End of a FOR-NEXT Loop	○	○	○

FNC 0 CJ	P	CJP (S)	Conditional Jump	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S													○			○

- M and VB series, S=P0 ~ P255
- VH series, S=P0 ~ P63



S : Destination Pointer of Conditional Jump

- When the conditional contact for the CJ instruction becomes "OFF"(CJ is not active), the program will keep running. When the conditional contact for the CJ instruction becomes "ON"(CJ is active), program will execute Jump actions and jump to the destination of CJ, and then keeps on running.
- When X20= "OFF", the CJ P15 instruction will execute Jump actions, and Program B will not be executed.
- When X20="ON", the CJ P0 instruction will execute Jump actions, and Program A will not be executed.
- If the CJ instruction is not executed, the program segment enclosed will be executed as normal programs.

When the CJ instruction executes Jump actions, every device of the skipped program segment will change as follows:

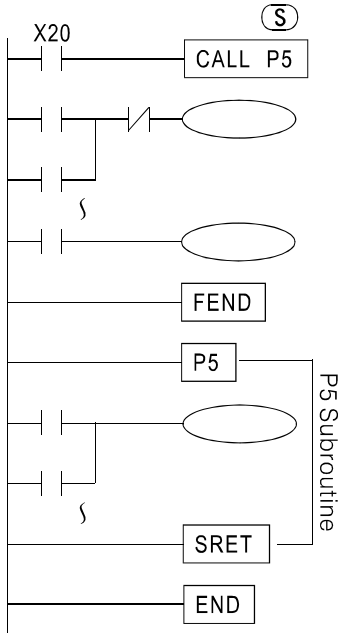
During Jump execution, the actions of every device in the program segment

- Y, M and S stay unchanged as before the Jump action.
- 10ms and 100ms Timers will stop counting time.
- 1ms Timer will continue to count time, but the output coil will not normally activate until the Jump stops.
- T192 ~ T199 will continue to count time and the output coil will also activate.
- High Speed Counter will continue to count and the output coil will also activate.
- Counter will stop counting.
- If the Reset instruction of Retentive timers and counters is driven before Jump, the device will still be reset during the Jump.
- Applied instructions will not be executed.

- Using the CJ instruction can skip unnecessary programs directly, so the program scan time can be saved.
- The CJ instruction can be used to solve the problem of double coil outputs.
- A Pointer numbered P can only appear once in a program; If the Pointer is specified more than once, errors will be incurred .
- As Pointer P255 is equal to the END address in a M or VB series program, CJ P255 is equal to jump to the END of a program.
- As Pointer P63 is equal to the END address in a VH series program, CJ P63 is equal to jump to the END of a program.

FNC 1 CALL	P	CALLP (S)	Call Subroutine	M	VB	VH
				○	○	○
FNC 2 SRET		SRET	Subroutine Return	M	VB	VH
				○	○	○

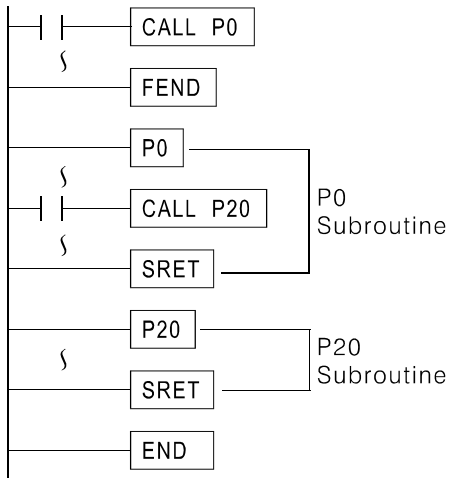
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S													○			○
• M and VB series, S = P0 ~ P254 • VH series, S = P0 ~ P62																



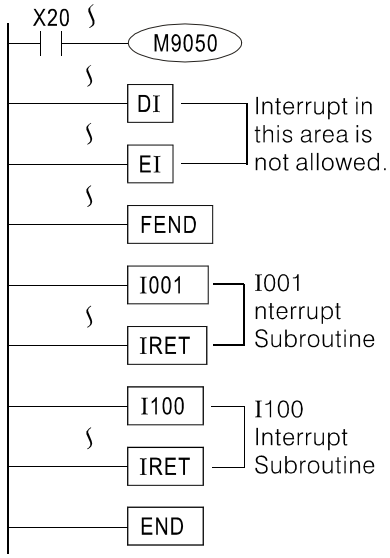
S : Subroutine Pointer

- When X20 = "ON", the CALL instruction will make the program flow jump to Pointer P5 to run subroutines, until an SRET instruction is encountered, where the program flow jumps back to the line of ladder logic immediately following the original CALL instruction and then keeps running.
- Subroutines should be written after the FEND instruction.
- If the CJ instruction and the CALL instruction are used in a program, the same Pointer number is not allowed.
- A same subroutine can be called in a program as many times as required.
- In a subroutine, a CALL instruction is available for calling other subroutines, while subroutines can be nested for 5 levels at most.
- The Timers used in the subroutine must be selected from the range T192 ~ T199 and T246 ~ T249. (VH series is not available).

- 2-Level Nest Subroutine Call (5 level at most)



	FNC 3 IRET		IRET	Interrupt Return	M	VB	VH
					○	○	○
	FNC 4 EI		EI	Enable Interrupt	M	VB	VH
					○	○	○
	FNC 5 DI		DI	Disable Interrupt	M	VB	VH
					○	○	○



- Generally a program is under Enable Interrupt status, but except the program flow is during the area between DI and EI, where the program is under Disable Interrupt.
- Assume that programs are under Enable Interrupt status: When X0= "OFF" → "ON", I001 Interrupt Subroutine will be executed until when the IRET instruction is encountered, then the flow returns to the main program and keep running. When X1= "ON" → "OFF ", I100 Interrupt Subroutine will be executed until when the IRET instruction is encountered, then the flow returns to the main program and keep running.
- When X20= "ON", the Interrupt Disable Special Coil M9050 is active and then I00□ is driven to disable Interrupt, the interrupt from the input terminal X0 is blocked.
- Please write Interrupt Pointer I after the FEND instruction.
- Generally, when the program flow executing an interrupt subroutine, all other interrupts are not allowed; But the EI and DI instructions interrupt subroutine can accept, this means that an interrupt subroutine may be interrupted during its operation, however at most 2 nested levels interrupt are accepted.

- The Timers used in general subroutines and interrupt subroutines must be selected from the range T192 ~ T199 and T246 ~ T249 (VH series is not available).
- When the program flow is worked between DI and EI, an interrupt demand cannot be executed immediately. The demand will be memorized, until the interrupt function is allowed, the interrupt subroutines will be executed.
- The pulse of the interrupt signal should be 200μs or longer.
- If the interrupt subroutine's I/O needs processed instantly, please use FNC53 immediate I/O refresh instruction.

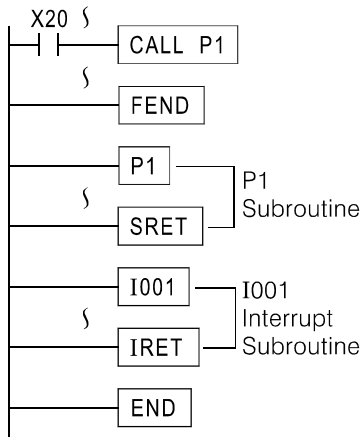
The assigned numbers for the Interrupt Pointer (I):

Input Interrupt		Timer Interrupt	High Speed Counter Interrupt
External Input Terminal	Interrupt Pointer	Interrupt Pointer	Interrupt Pointer
X0	I00 □	3 points: I6 □□ I7 □□ I8 □□	6 Points: I010 I020 I030 I040 I050 I060
X1	I10 □		
X2	I20 □		
X3	I30 □		
X4	I40 □		
X5	I50 □		
□=1, indicates the interrupt during rising □=0, indicates the interrupt during falling		□□=01 ~ 99, indicate Timer Interrupt interval length, where the time interval will be 1 ~ 99 ms	With the instruction FNC53 (DHSCS) to make a interrupt signal

Interrupt Control Special Coils:

Coil ID No.	Instruction of Function
M9050	Input interrupt I00□ is prevented.
M9051	Input interrupt I10□ is prevented.
M9052	Input interrupt I20□ is prevented.
M9053	Input interrupt I30□ is prevented.
M9054	Input interrupt I40□ is prevented.
M9055	Input interrupt I50□ is prevented.
M9056	Timer interrupt I6□□ is prevented.
M9057	Timer interrupt I7□□ is prevented.
M9058	Timer interrupt I8□□ is prevented.
M9059	High Speed Counter interrupt I010 ~ I060 is prevented.

FNC 6 FEND	FEND	First End	M	VB	VH
			○	○	○



- An FEND instruction indicates the first end of a main program block.
- An FEND instruction placed before CALL instruction or after SRET instruction will be deemed as an error.
- Pointer P and Interrupt Pointer I which is specified by CALL instruction should be written behind the FEND instruction.
- If two or more FEND instructions are used, the subroutine should be placed between the last FEND instruction and END instruction.

FNC 7 WDT	P	WDTP	Watch Dog Timer Refresh	M	VB	VH
				○	○	○

PLC is provided with a WTC (Watch Dog Timer), which is used to monitor operation condition of the PLC system. If any trouble occurs to PLC's CPU, through the WDT's monitoring, will command PLC to stop operation and turn all external output "OFF" to achieve the protection purpose.

The WDT (Watch Dog Timer) action statement:

WDT is a hardware timer (a 200ms timer, because when PLC= "STOP" → "RUN", the value of WDT will reload from the content value of Special Register D9000, while the setting value of D9000 is "200") counting time downward by a timing unit of 1ms. If the value reaches i0i, WDT will determine that there is a system trouble, it forces the PLC to stop operation and turn all external output "OFF" to achieve the protection purpose. When the system operates normally, PLC will revert its WDT timer before it executes the beginning of program (STEP 0).

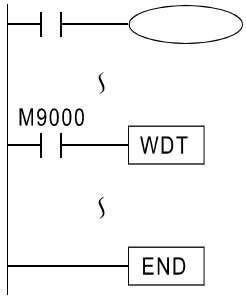
There are two reasons to activate WTD (Watch Dog Timer) function:

(1) Any trouble is happened in the PLC system and WDT performs the protection function.

(2) If the time of program execution is too long, the program's scan time more than the content value of D9000, it will triggers the protection function of WDT. Below are two approaches to improve the foregoing situation and make the system operate normally.

triggers the protection function of WDT. Below are two approaches to improve the foregoing situation and make the system operate normally.

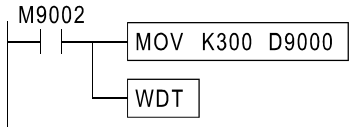
① Insert WDT instruction into the program, because WDT instruction will revert the timing value of WDT.



② Use MOV instruction to change the content value of D9000.

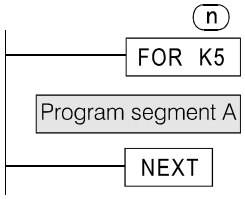
Setting WDT timer as 300ms

To adopt this approach, it should be noted that on the first scan time of PLC="STOP" → "RUN" , the value of WDT timer is still 200ms. The program below can be used for the solution where necessary.



FNC 8 FOR		FOR (n)	Start of a FOR-NEXT Loop	M	VB	VH
				○	○	○
FNC 9 NEXT		NEXT	End of a FOR-NEXT Loop	M	VB	VH
				○	○	○

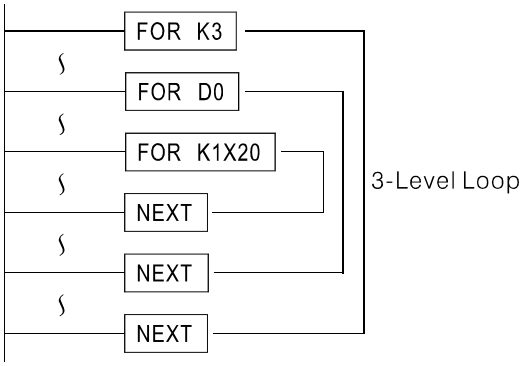
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
n					○	○	○	○	○	○	○	○		○	○	○
• n =1 ~ 32,767 (Otherwise, n=1, if the setting value exceeds beyond the range)																



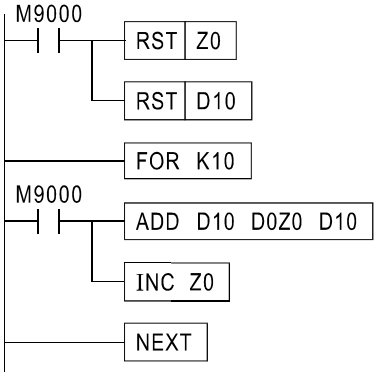
n : The number of times to be repeated in FOR-NEXT loop.

- The program in the FOR-NEXT loop will be executed "n" times.
- As in the left diagram, Program segment A is executed 5 times sequentially.

- In a For-Next loop, CJ instruction can be used to jump out of the loop.
- At most 5 levels can be used for a next FOR-NEXT loop. Be sure to note that the loop should be taken not to exceed WDT's default value, otherwise an error will occur.
- Errors will occur under the following circumstances:
  NEXT instruction is placed in front of FOR instruction.
  NEXT instruction is placed behind FEND or END instruction.
  FOR instruction and NEXT instruction are not programmed as a pair.
- Multiple-level Loop Program



- Using FOR-NEXT Loop instructions jointly with Pointer Register V, Z will make programs more flexible. The program below will add up the content value of D0 ~ D9 and store the result in D10.



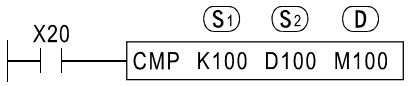
MEMO

6-3 Compare and Transfer Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
10	D	CMP	P	Compare	○	○	○
11	D	ZCP	P	Zone Compare	○	○	○
12	D	MOV	P	Move	○	○	○
13		SMOV	P	Shift Move	○	○	○
14	D	CML	P	Compliment	○	○	○
15		BMOV	P	n→n Block Move	○	○	○
16	D	FMOV	P	1→n Fill Move	○	○	○
17	D	XCH	P	Exchange	○	○	○
18	D	BCD	P	Converts BIN to BCD	○	○	○
19	D	BIN	P	Converts BCD to BIN	○	○	○

D	FNC 10 CMP	P	DCMP(P) (S1) (S2) (D)	Compare	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
D		○	○	○												○
• D occupies 3 consecutive devices																



S1 : Compare Value 1

S2 : Compare Value 2

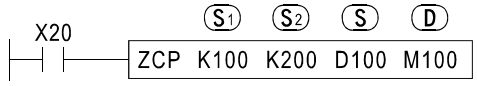
D : Compare Result; occupying 3 consecutive points

- Compare the data of (S1) with the data of (S2) and save the result in (D) (Compare Result).
- The CMP instruction will be enabled when X20= "ON".
  If K100>D100, then M100= "ON" ;
  If K100=D100, then M101= "ON" ;
  If K100<D100, then M102= "ON" .
- When X20= "OFF", the instruction is disabled, the status ("ON"/ "OFF") of M100, M101 and M102 remains as the status before X20= "OFF".
- Please use serial or parallel links of M100 ~ M102 to generate the result as "≥", "≤" or "≠".

D	FNC 11 ZCP	P	DZCP P (S1) (S2) (S) (D)	Zone Compare	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
S					○	○	○	○	○	○	○	○		○	○	○
D		○	○	○												○

- D occupies 3 consecutive devices
- S1 ≤ S2

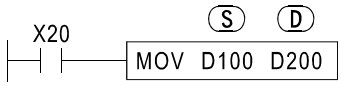


S1 : Lower limit of zone compare
S2 : Upper limit of zone compare
S : Compare Value
D : Compare Result; occupying 3 consecutive points

- Compare the data of (S) with the data of (S1), the data of (S2), and save the result in (D) (Compare Result).
- The CMP instruction will be enabled when X20= "ON".
  If K100>D100 (Lower Limit>Compare Value), then M100= "ON";
  If K100≤D100≤K200 (Compare Value is located between Upper Limit and Lower Limit), then M101= "ON";
  If K200<D100 (Compare Value>Upper Limit), then M102= "ON".
- The instructions is disabled when X20= "OFF". The status ("ON"/ "OFF") of M100, M101 and M102 remains as the status before X20= "OFF".
- When (S1) > (S2), the value of (S1) will become both of the Upper/Lower Limits to compare with (S).

D	FNC 12 MOV	P	DMOVP (S) (D)	Move	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

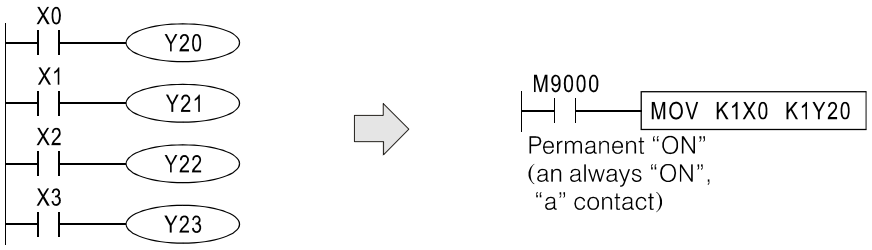


S : Source Device of Transfer
D : Destination Device

- To copy the designated value from (S) to (D).
- The content value of D100 will be copied to D200 when X20= "ON".
- The instruction is disabled and D200 remains invariable when X20= "OFF".

Bit Transfer

To perform the program of left diagram, which can be changed as the right side.

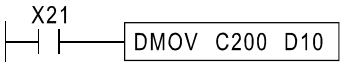


32-bit Data Transfer

The instruction should be headed with a "D" when a 32-bit instruction is used.

When X20= "ON", move the content value (D1, D0) to (D101, D100).

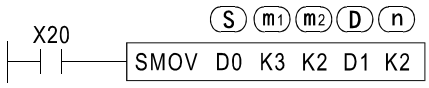
If the transfer target is a 32-bit counter, the instruction should be headed with a "D".



When X21= "ON", move the current value of C200 (32 bits) to (D11, D10).

FNC 13 SMOV	P	SMOVP (S) (m1) (m2) (D) (n)	Shift Move	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○		○
m1															○	
m2															○	
D									○	○	○	○		○		○
n															○	
• m1=1~4; m2=1~m1; n=m2~4																



S : Source Device of Transfer

m1 : The source position of the first digit to be moved

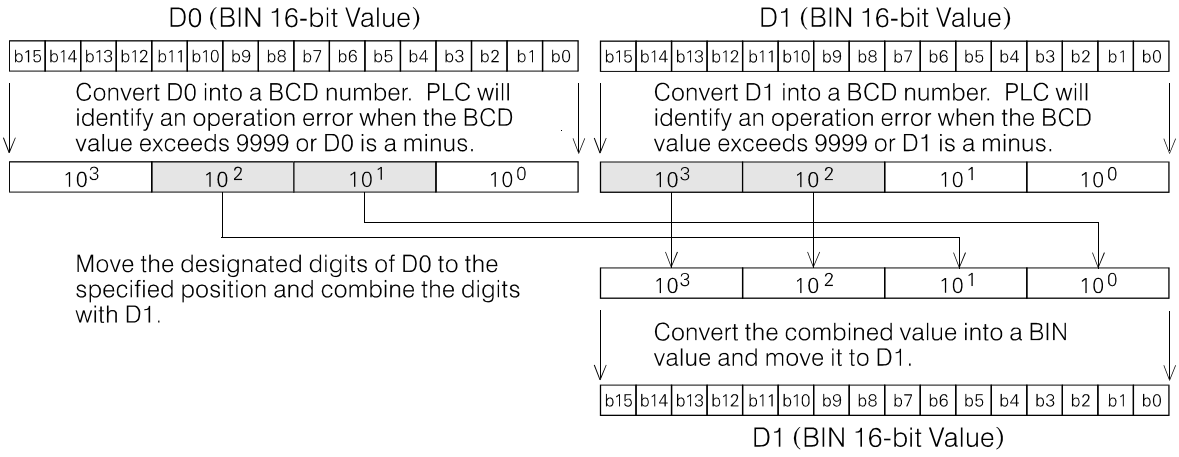
m2 : The number of source digits to be moved

D : Destination Device

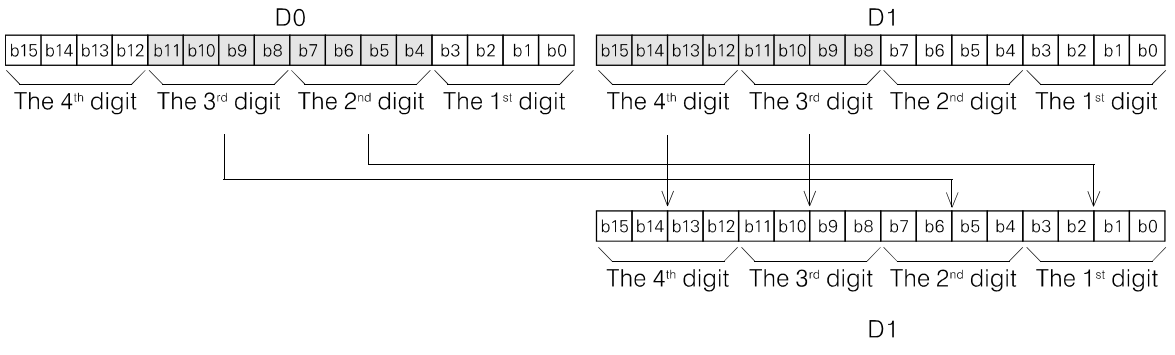
n : Destination position for the first digit

- This instruction can be used for data reorganization.
- The instruction can select different operation modes, it is based on the status of Special Coil M9168.

When M9168= "OFF"

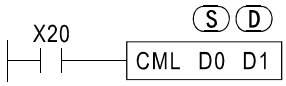


When M9168= "ON"



D	FNC 14 CML	P	DCMLP (S) (D)	Compliment	M	VB	VH
					○	○	○

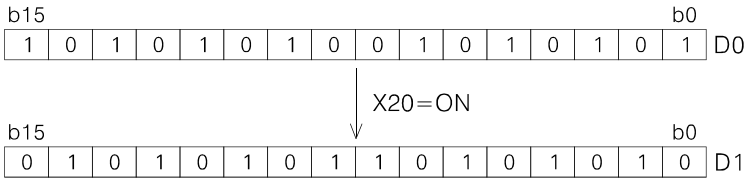
Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○



S : Source Device of Transfer

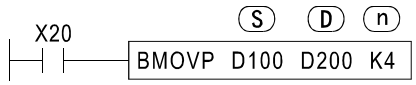
D : Destination Device

- Invert all contents of (S) (i.e. “0” is inverted as”1” and “1”, inverted as “0”, for each digit) and copy the contents to (D) .
- When X20= “ON”, all of contents of D0 are inverted and copied to D1.
- When X20= “OFF”, the instruction is disabled and the contents of D1 remains invariable.

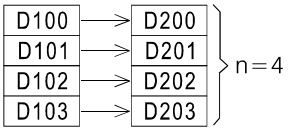


FNC 15 BMOV	P	BMOVP (S) (D) (n)	n→n Block Move	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D						○	○	○	○	○	○					○
n															○	
• n=1～512																

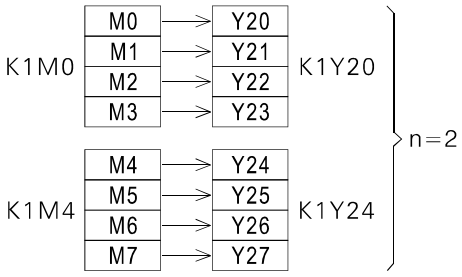


S : The head ID number of Source Device
D : The head ID number of Destination Device
n : Length of the block to be moved



- BMOV executes (S) → (D) "n" consecutive points of a data transfer.
- When X20= "OFF" → "ON", the content value of D100～D103 will be moved to D200～D203 orderly.

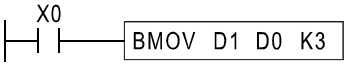
- When a block transfer of bit devices is executed, the data ranges of (S) and (D) should coincide.

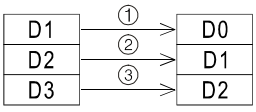


- When X20= "ON", K1M0 and K1M4 (equal to M0～M7) will be copied to K1Y20 and K1Y24 (equal to Y20～Y27).

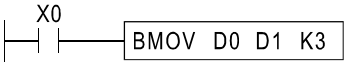
- To prevent data writing errors during the transfer, the transfer will be processed in different orders when (S)>(D) or (S)<(D).

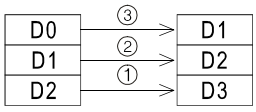
The transfer order when (S)>(D)





The transfer order when (S)<(D)

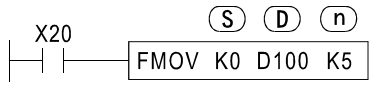




- Read and Write of File Register are required to be completed with the BMOV instruction. Please refer to Section 2-9 "File Register" for details.

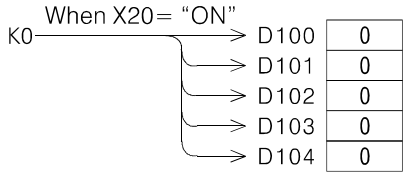
D	FNC 16 FMOV	P	DFMOVP (S) (D) (n)	1→n Fill Move	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○					○
n															○	
• n=1 ~ 512																



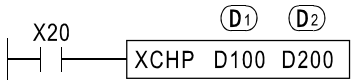
S : Source Device of Transfer
D : The head ID number of Destination Device
n : Length of the block to be moved

- Move the content value of (S) to (n) registers which headed with (D).
- When X20= "ON", K0 will be copied to 5 continuous registers headed with D100 (D100 ~ D104).
- If the range designated by (n) which is exceed the available devices space at the destination location, then only the available destination devices will be copied to.



D	FNC 17 XCH	P	DXCHP (D1) (D2)	1→n Fill Move	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D1						○	○	○	○	○	○	○		○		○
D2						○	○	○	○	○	○	○		○		○



D1: Data 1 to be exchanged
D2: Data 2 to be exchanged

- Exchange (swap) contents values of the devices (D1) and (D2).
- When X20="OFF" → "ON", content values of (D100) and (D200) will be exchanged.

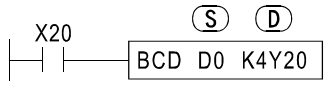
Before Execution			After Execution	
123	D100	When X20= "OFF" → "ON"	60	D100
60	D200		123	D200

- When X20="OFF" → "ON", content values of (D100) and (D200) will be exchanged.

Before Execution			After Execution	
5	D0		15	D0
10	D1	When X21= "OFF" → "ON"	20	D1
15	D100		5	D100
20	D101		10	D101

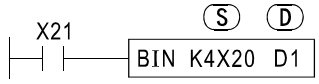
					M	VB	VH
D	FNC 18 BCD	P	DBCDP (S) (D)	Converts BIN to BCD	○	○	○
D	FNC 19 BIN	P	DBINP (S) (D)	Converts BCD to BIN	○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○		○
D						○	○	○	○	○	○	○		○		○



S : Converted Source (BIN)
D : Converted Result Destination (BCD)

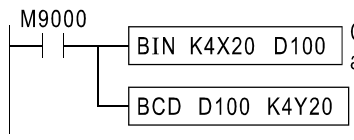
- When X20= "ON", the BIN value in D0 will be converted into a BCD value. And then, moved to K4Y20 (Y20 ~ Y37).
- For a 16-bit instruction, PLC will identify an error when (S) exceeds the operational range (0 ~ 9,999).
- For a 32-bit instruction, PLC will identify an error when (S) exceeds the operational range (0 ~ 99,999,999).



S : Converted Source (BCD)
D : Converted Result Destination (BIN)

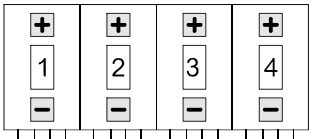
- When X21= "ON", the BCD value in K4X20 (X20 ~ X37) will be converted into a BIN value. And then, moved to D1.
- If the Source data is not provided in a BCD format, PLC will identify an operation error.

Application of BCD and BIN Instructions

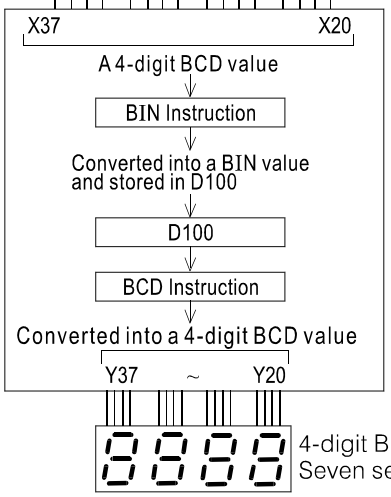


Convert the BCD value of K4X20 into an equivalent BIN value and move it to D100.

Convert the BIN value of D100 into an equivalent BCD value and move it to K4X20.



4-digit BCD-based Thumbwheel Switch



4-digit BCD-based Seven segment display

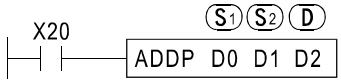
- Data in PLC are all stored in a BIN format, and applied instructions (arithmetic and logical operations, ect.) are also executed based on BIN values.
- When a PLC is reading a BCD-based thumbwheel switches, which is required to use the BIN instruction to convert the data into a BIN value and store it in the PLC.
- If a PLC is used to output inner stored data to a seven segment display (BCD format), please use the BCD instruction to convert inner data into a BCD value and move it to the display.

6-4 Arithmetic and Logical Operations

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
20	D	ADD	P	Addition (S1)+(S2) → (D)	○	○	○
21	D	SUB	P	Subtraction (S1) – (S2) → (D)	○	○	○
22	D	MUL	P	Multiplication (S1)×(S2) → (D+1,D)	○	○	○
23	D	DIV	P	Division (S1)÷(S2) → (D),(D+1)	○	○	○
24	D	INC	P	Increment (D)+1 → (D)	○	○	○
25	D	DEC	P	Decrement (D) – 1 → (D)	○	○	○
26	D	WAND	P	Logic Word AND (S1)∧(S2) → (D)	○	○	○
27	D	WOR	P	Logic Word OR (S1)∨(S2) → (D)	○	○	○
28	D	WXOR	P	Logic Word exclusive OR (S1)⩛(S2) → (D)	○	○	○
29	D	NEG	P	Negation $(\overline{D})+1$ → (D)	○	○	

D	FNC 20 ADD	P	DADDP (S1) (S2) (D)	Addition (S1)+(S2) → (D)	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

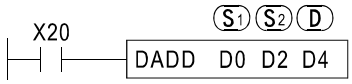


S1 : Summand
S2 : Addend
D : Total

- When X20= "OFF" → "ON" , the summand (D0) will be added to the addend (D1), and the total will be stored at the specified destination device (D2).

```
  10   D0
+  5   D1
------
  15   D2
```

- 16-bit Operation
  When the result of an operation, (D) , is equal to "0", the zero flag M9020= "ON".
  When the result of an operation exceeds 32,767, the carry flag M9022= "ON".
  When the result of an operation is less than -32,768, the borrow flag M9021= "ON".



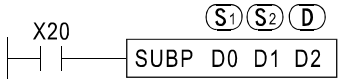
- When X20= "ON", add (D1, D0) and (D3, D2) together and store the total in (D5, D4).

```
 100,000  (D1,D0)
+  -100   (D3,D2)
---------
  99,900  (D5,D4)
```

- 32-bit Operation
  When the result of an operation, (D) , is equal to "0", the zero flag M9020= "ON".
  When the result of an operation exceeds 2,147,483,647, the carry flag M9022= "ON".
  When the result of an operation is less than -2,147,483,648, the borrow flag M9021= "ON".

D	FNC 21 SUB	P	DSUBP (S1) (S2) (D)	Subtraction (S1) – (S2) → (D)	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

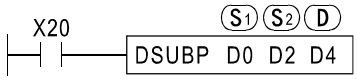


S1 : Minuend
S2 : Subtrahend
D : Remainder

- When X20= “OFF” → “ON” , the subtrahend (D1) will be subtracted from the minuend (D0), and the remainder will be stored at the destination device (D2).

```
  10   D0
– 5    D1
-------
  5    D2
```

- 16-bit Operation
  When the result of an operation, (D), is equal to “0”, the zero flag M9020= “ON”.
  When the result of an operation exceeds 32,767, the carry flag M9022= “ON”.
  When the result of an operation is less than -32,768, the borrow flag M9021= “ON”.



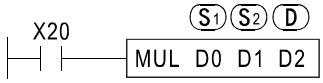
- When X20="ON", subtract (D3, D2) from (D1, D0) and store the remainder in (D5, D4).

```
  100,000  (D1,D0)
– – 100    (D3,D2)
-----------
  100,100  (D5,D4)
```

- 32-bit Operation
  When the result of an operation, (D), is equal to “0”, the zero flag M9020= “ON”.
  When the result of an operation exceeds 2,147,483,647, the carry flag M9022= “ON”.
  When the result of an operation is less than –2,147,483,648, the borrow flag M9021= “ON”.

D	FNC 22 MUL	P	DMULP (S1) (S2) (D)	Multiplication (S1)×(S2) → (D+1,D)	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
D											○					○

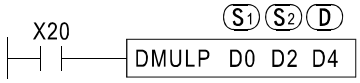


S1 : Multiplicand
S2 : Multiplier
D : Product (of a multiplication)

- When X20= "ON", the multiplicand (D0) will be multiplied by the multiplier (D1), and the remainder will be stored at the destination device (D3, D2).

	10	D0
×	5	D1
	50	(D3,D2)

- Two 16-bit data sources multiplied together will create a 32-bit product.
- The Most Significant Bit (MSB) of a 32-bit product indicates a positive or negative ("0" represents a positive and "1" represents a negative).



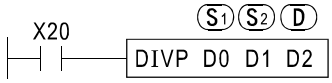
- When X20= "ON", multiply (D1, D0) by (D3, D2) and store the product in (D7, D6, D5, D4).

	100,000	(D1,D0)
×	−10	(D3,D2)
	−1,000,000	(D7,D6,D5,D4)

- A 32-bit multiplicand multiplied by a 32-bit multiplier will create a 64-bit product.
- The Most Significant Bit (MSB) of a 64-bit product indicates a positive or negative ("0" represents a positive and "1" represents a negative).

D	FNC 23 DIV	P	DDIVP (S1) (S2) (D)	Division (S1)÷(S2) → (D),(D+1)	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
D											○					○

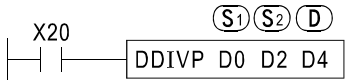


S1 : Dividend
S2 : Divisor
D : Quotient and Remainder

- When X20= "OFF" → "ON" , the dividend (D0) will be divided by the divisor (D1), and the quotient will be stored at the destination device (D2) while the remainder will be stored in (D3).

	10	D0
÷	−3	D1
Quotient	−3	D2
Remainder	1	D3

- In case a 16-bit quotient and a 16-bit remainder are created, the Most Significant Bit will indicate a positive or negative ("0" represents a positive and "1" represents a negative).



- When X20= "OFF" → "ON" , divide (D1, D0) by (D3, D2) and store the quotient in (D5, D4), store the remainder in (D7, D6).

	−300	(D1,D0)
÷	−11	(D3,D2)
Quotient	27	(D5,D4)
Remainder	−3	(D7,D6)

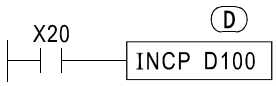
- In case a 32-bit quotient and a 32-bit remainder are yielded, the most significant bit will indicate a positive or negative ("0" represents a positive and "1" represents a negative).

Note:

- PLC will identify an operation error, if the divisor is equal to "0".
- The quotient of a positive dividend and a positive divisor (or a negative dividend and a negative divisor) will automatically be a positive; If either of a dividend or divisor is positive and the other is negative, the quotient will automatically be a negative.
- A positive dividend produces a positive remainder, while a negative dividend produces a negative remainder.

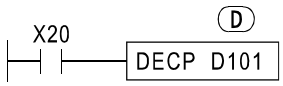
					M	VB	VH
D	FNC 24 INC	P	DINCP (D)	Increment (D)+1 → (D)	○	○	○
D	FNC 25 DEC	P	DDECP (D)	Decrement (D) – 1 → (D)	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D						○	○	○	○	○	○	○		○		○



D : Destination Device

- When X20= “OFF” → “ON” , the current value of destination (D100) will have its value incremented (increased) by a value of “1”.
- If the instruction is not a pulse (P) instruction, (D100) will have its value incremented by a value of “1” in every scan cycle.
- In a 16-bit operation, when a value of “+32,767” is reached, the next increment of “1” will write a value of “-32,768” to the destination device.
- In a 32-bit operation, when a value of “+2,147,483,647” is reached, the next increment of “1” will write a value of “-2,147,483,648” to the destination device.
- The instruction operation result will never lead to any change of a flag.

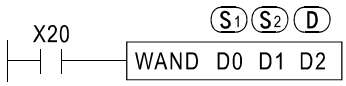


D : Destination Device

- When X20= “OFF” → “ON” ,the current value of destination (D101) will have its value decremented (decreased) by a value of “1”.
- If the instruction is not a pulse (P) instruction, (D101) will have its value decremented by a value of “1” in every scan cycle.
- In a 16-bit operation, when a value of “-32,768” is reached, the next decrement of “1” will write a value of “+32,767” to the destination device.
- In a 32-bit operation, when a value of “-2,147,483,648” is reached, the next increment of “1” will write a value of “+2,147,483,647” to the destination device.
- The instruction operation result will never lead to any change of a flag.

					M	VB	VH
D	FNC 26 WAND	P	DWANDP (S1) (S2) (D)	Logic Word AND (S1)^(S2) → (D)	○	○	○
D	FNC 27 WOR	P	DWORP (S1) (S2) (D)	Logic Word OR (S1)∨(S2) → (D)	○	○	○
D	FNC 28 WXOR	P	DWXORP (S1) (S2) (D)	Logic Word exclusive OR (S1)⩛(S2) → (D)	○	○	○

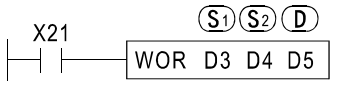
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○



S1 : Source Device 1
S2 : Source Device 2
D : Operation Result

- When X20= “ON”, 16 bits of (D0) and (D1) execute the logic AND operation and restore the result in (D2).
- The logic AND operation rules are: 0^0=0, 0^1=0, 1^0=0 and 1^1=1; any “0” will cause a result of “0”.

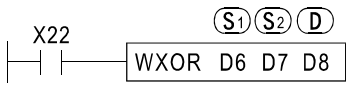
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	D0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	D1
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	D2



S1 : Source Device 1
S2 : Source Device 2
D : Operation Result

- When X20= “ON”, 16 bits of (D3) and (D4) execute the logic OR operation, and restore the result in (D5).
- The logic OR operation rules are: 0∨0=0, 0∨1=0, 1∨0=0 and 1∨1=1; any “1” will cause a result of “1”.

0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	D3
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	D4
0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	D5



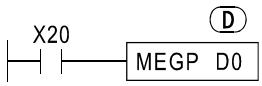
S1 : Source Device 1
S2 : Source Device 2
D : Operation Result

- When X20= “ON”, 16 bits of (D6) and (D7) execute the logic XOR operation, and restore the result in (D8).
- The logic XOR operation rules are: 0⩛0=0, 0⩛1=1, 1⩛0=1 and 1⩛1=0; same values will cause a result of “0”, otherwise, “1”.

0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	D6
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	D7
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	D8

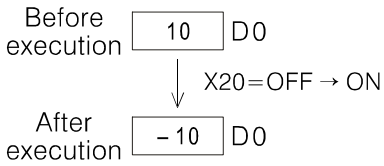
D	FNC 29 NEG	P	DNEGP (D)	Negation $(\overline{D})+1\rightarrow(D)$	M	VB	VH
					○	○	

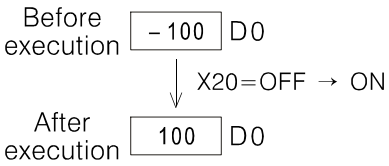
Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
D						○	○	○	○	○	○	○		○		○



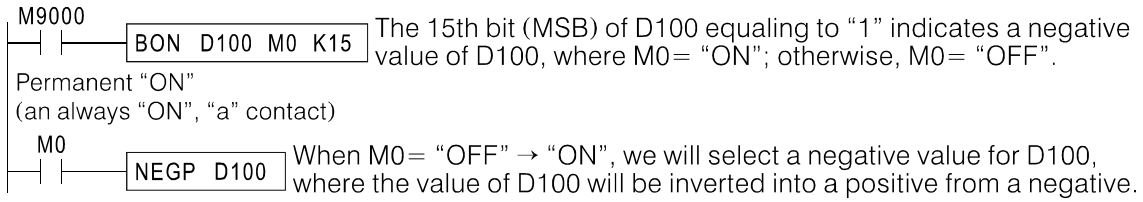
D : the selected device to be inverted

- When X20= “OFF” → “ON” , each single bit pattern of (D0) will be inverted (“0” inverted into “1” and vice versa) and then added with “1”. The result will be stored in (D0). The instruction select the complement of “2” for the value of .(D) The operation changes the positive or negative symbol of a value. For example,





- The absolute value of D100 can be generated with the following program.



The 15th bit (MSB) of D100 equaling to “1” indicates a negative value of D100, where M0= “ON”; otherwise, M0= “OFF”.

When M0= “OFF” → “ON”, we will select a negative value for D100, where the value of D100 will be inverted into a positive from a negative.

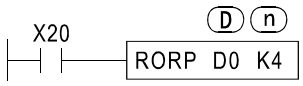
6-5 Rotary and Shift Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
30	D	ROR	P	Rotation Right	○	○	○
31	D	ROL	P	Rotation Left	○	○	○
32	D	RCR	P	Rotation Right with Carry	○	○	○
33	D	RCL	P	Rotation Left with Carry	○	○	○
34		SFTR	P	Bit Shift Right	○	○	○
35		SFTL	P	Bit Shift Left	○	○	○
36		WSFR	P	Word Shift Right	○	○	
37		WSFL	P	Word Shift Left	○	○	
38		SFWR	P	Shift Register Write (FIFO Write)	○	○	○
39		SFRD	P	Shift Register Read (FIFO Read)	○	○	○

					M	VB	VH
D	FNC 30 ROR	P	DRORP (D) (n)	Rotation Right	○	○	○
D	FNC 31 ROL	P	DROLP (D) (n)	Rotation Left	○	○	○

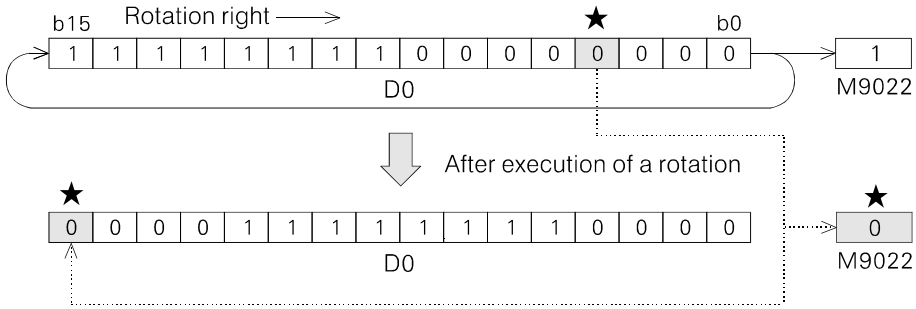
Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
D						○	○	○	○	○	○	○		○		○
n															○	

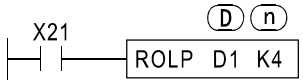
- The 16-bit instruction n=1 ~ 16
- The 32-bit instruction n=1 ~ 32
- When D is designated as KnY, KnM and KnS, the 16-bit instruction can only designate K4Y, K4M and K4S, while the 32-bit instruction can only designate K8Y, K8M and K8S



D : the selected device to be rotated
n : number of the bits to be rotated

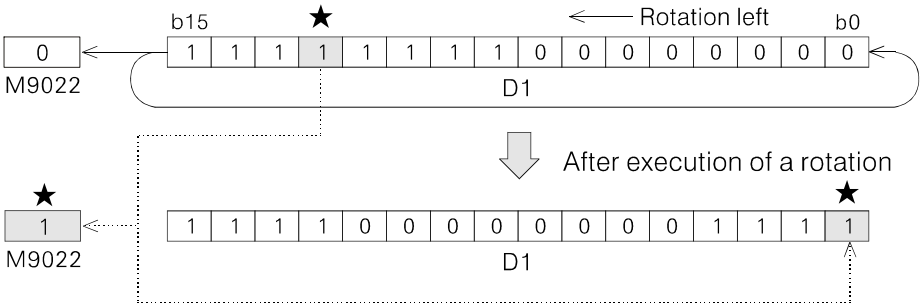
- The bit pattern of the device designated by (D) is rotated (n) bit places to the right.
- When X20= "OFF" → "ON", the 16-bit data of (D0) will be rotated 4 bits to the right, and the status of the rotated data will be copied to the carry flag M9022.





D : the selected device to be rotated
n : number of the bits to be rotated

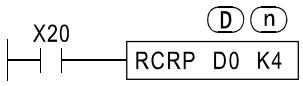
- The bit pattern of the device designated by (D) is rotated (n) bit places to the left.
- When X21= "OFF" → "ON", the 16-bit data of (D1) will be rotated 4 bits to the left, and the status of the rotated data will be copied to the carry flag M9022.



					M	VB	VH
D	FNC 32 RCR	P	DRCRP (D) (n)	Rotation Right with Carry	○	○	○
D	FNC 33 RCL	P	DRCLP (D) (n)	Rotation Left with Carry	○	○	○

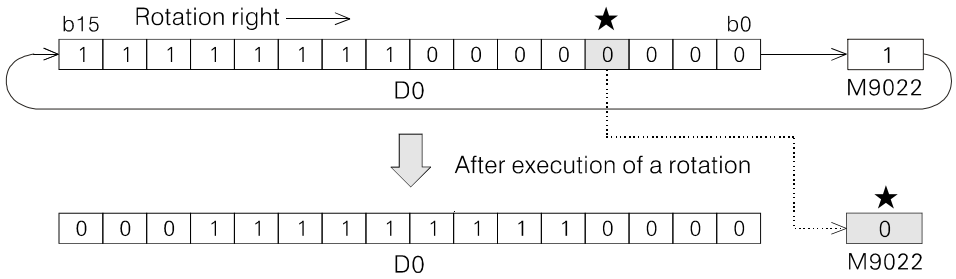
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D						○	○	○	○	○	○	○		○		○
n															○	

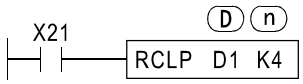
- The 16-bit instruction n=1 ~ 16
- The 32-bit instruction n=1 ~ 32
- When D is designated as K*n*Y, K*n*M and K*n*S, the 16-bit instruction can only designate K4Y, K4M and K4S, while the 32-bit instruction can only designate K8Y, K8M and K8S



D : the selected device to be rotated
n : number of the bits to be rotated

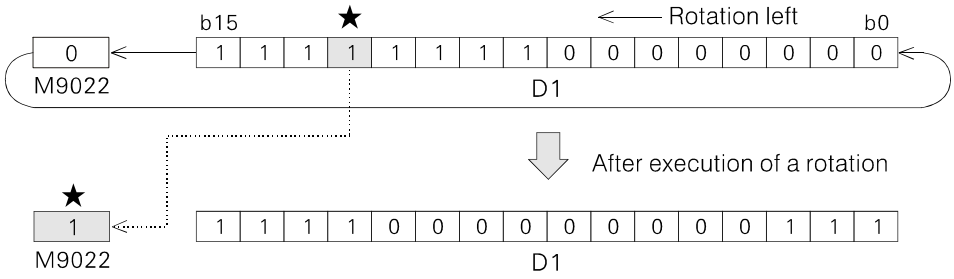
- The contents of the device designated by (D) are rotated "n" bit places to the right with (n) bits extracted from the carry flag M9022.
- When X20= "OFF" → "ON", the 16-bit data of (D0) will be rotated 4 bits to the right with 4 bits extracted from the carry flag M9022.





D : the selected device to be rotated
n : number of the bits to be rotated

- The contents of the device designated by (D) are rotated "n" bit places to the left with (n) bits extracted from the carry flag M9022.
- When X21= "OFF" → "ON", the 16-bit data of (D1) will be rotated 4 bits to the left with 4 bits extracted from the carry flag M9022.



FNC 34 SFTR	P	SFTRP (S) (D) (n1) (n2)	Bit Shift Right	M	VB	VH
				○	○	○
FNC 35 SFTL	P	SFTLP (S) (D) (n1) (n2)	Bit Shift Left	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S	○	○	○	○												○
D		○	○	○												○
n1															○	
n2															○	

● n1=1 ~ 1024 ● n2=1 ~ n1

```
X20
 ┤ ├──[ SFTRP  X0  M0  K16  K4 ]
```
(S) (D) (n1) (n2)

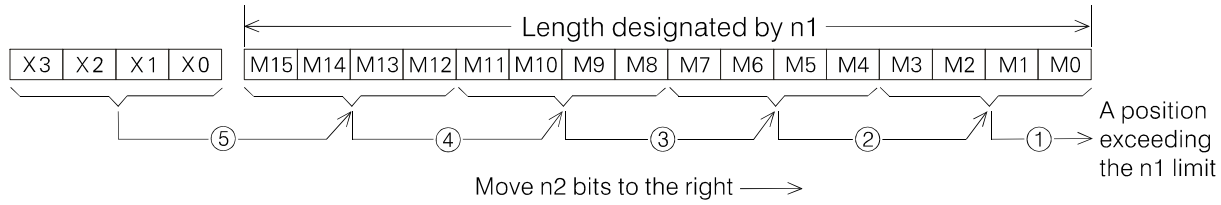
S : The head of source device ID number to be Moved in

D : The head of destination device ID number to be shifted

n1: data length to be shifted

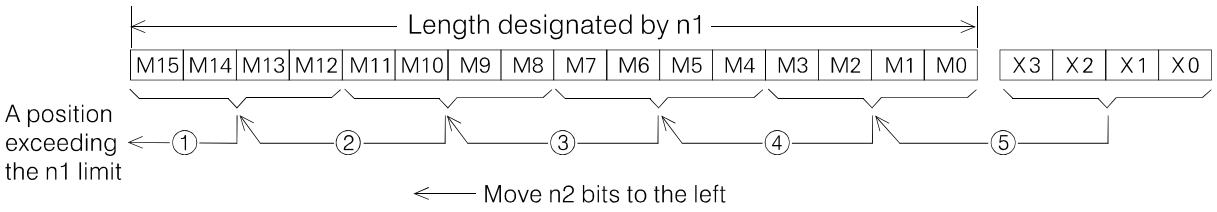
n2: number of the bits in a shift

- Move the length of (n1) bits of a device, headed with (D), (n2) bits to the right. A device headed with (S) will be used as the output complementary bit during the shift.
- When X20= "OFF" → "ON",the device composed of M0 ~ M15 (16 bits) will be moved 4 bits to the right; X0 ~ X3 will be moved in M12 ~ M15 for use of the output complementary bits.




```
X21
 ┤ ├──[ SFTLP  X0  M0  K16  K4 ]
```
(S) (D) (n1) (n2)

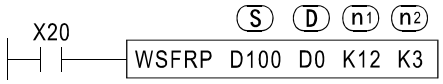
- Move the length of (n1) bits of a device, headed with (D), (n2) bits to the left. A device headed with (S) will be used as the output complementary bit during the shift.
- When X21= "OFF" → "ON",the device composed of M0 ~ M15 (16 bits) will be moved 4 bits to the left; X0 ~ X3 will be moved in M0 ~ M3 for use of the output complementary bits.



FNC 36 WSFR	P	WSFRP (S) (D) (n1) (n2)	Word Shift Right	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D						○	○	○	○	○	○					○
n1															○	
n2															○	

- n_1=1 ~ 512
- n_2=1 ~ n_1



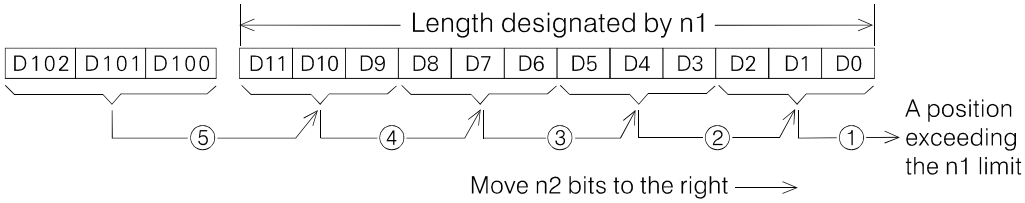
S : The head of source device ID number to be Moved in

D : The head of destination device ID number to be shifted

n1: data length to be shifted

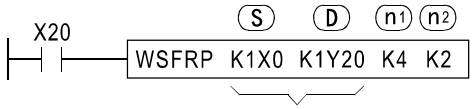
n2: number of the word in a shift

- Move a word stack with the length of (n1) words of a device, headed with (D), (n2) words to the right. A device headed with (S) will be used as the output complementary word during the shift.
- When X20= "OFF" → "ON",the word stack composed of D0 ~ D11 (12 words) will be moved 3 words to the right; D100 ~ D102 will be moved in D9 ~ D11 for use of the output complementary words.

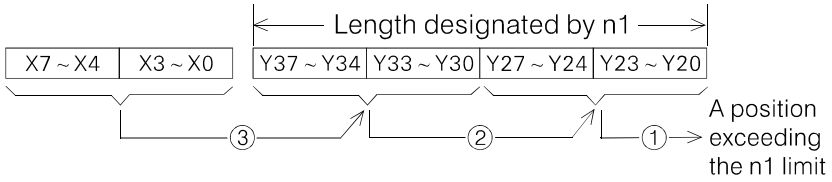


Note:

- The device properties designated by (S) and (D) must be the same (both are word devices or bit devices).
- When (S) and (D) designate bit devices, the digits of K*n* must be the same.



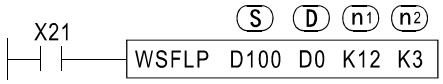
The digits designated by (S) and (D) must be the same.



FNC 37 WSFL	P	WSFLP (S) (D) (n1) (n2)	Word Shift Left	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D						○	○	○	○	○	○					○
n1															○	
n2															○	

● n1=1 ~ 512 ● n2=1 ~ n1



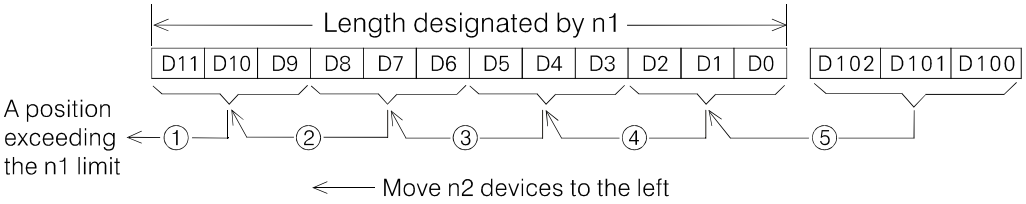
S : The head of source device ID number to be Moved in

D : The head of destination device ID number to be shifted

n1: data length to be shifted

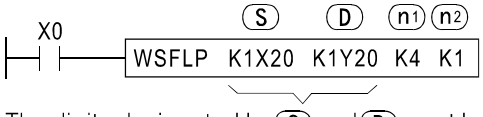
n2: number of the word in a shift

- Move a word stack with the length of (n1) words of a device, headed with (D), (n2) words to the left. A device headed with (S) will be used as the output complementary word during the shift.
- When X21 = "OFF" → "ON", the word stack composed of D0 ~ D11 (12 words) will be moved 3 words to the left; D100 ~ D102 will be moved in D0 ~ D2 for use of the output complementary words.

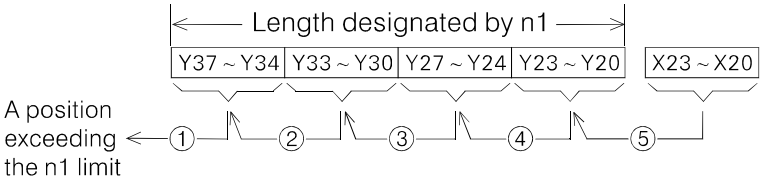


Note:

- The device properties designated by (S) and (D) must be the same (both are word devices or bit devices).
- When (S) and (D) designate bit devices, the digits of K*n* must be the same.

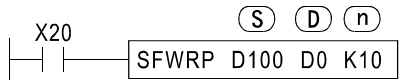


The digits designated by (S) and (D) must be the same.



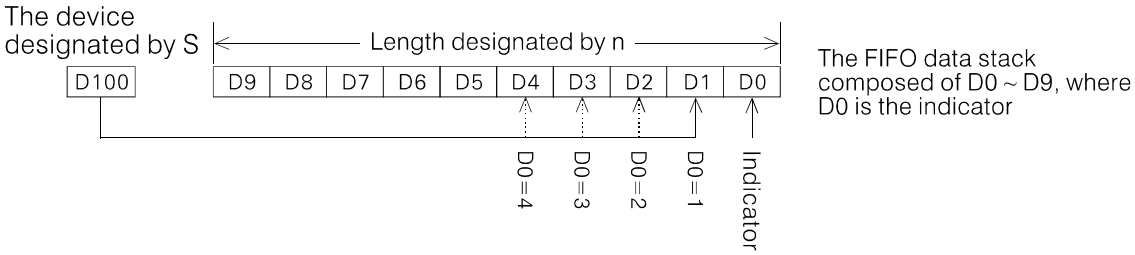
FNC 38 SFWR	P	SFWRP (S) (D) (n)	Shift Register Write (FIFO Write)	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D									○	○	○					○
n															○	
• n=2～512																



S : the device to be written to a FIFO data stack
D : source digit number of the FIFO data stack
n : Length of the FIFO data stack

- The data stack of (n) words, headed with (D), is defined as the FIFO data stack. The first device of the FIFO data stack is designated as the indicator. When the instruction is enabled, the content value of the indicator will be added with "1" firstly, and then, the content value of the device designated by will be moved to the position, designated by (S) the indicator, in the FIFO data stack.



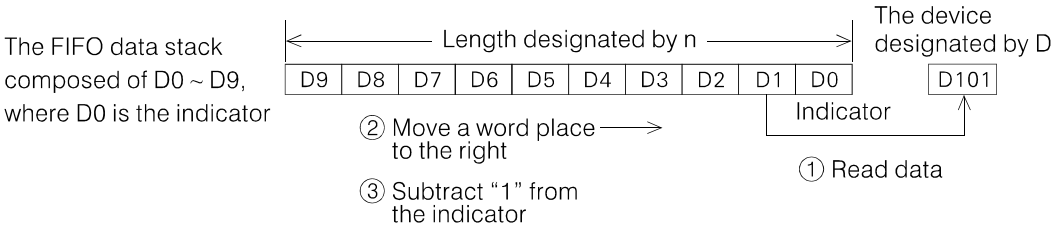
- Suppose (D0)=0. When X20= "OFF" → "ON", the content value of (D0) will become "1" and the content value of (D100) will be moved to (D1). If, again, X20= "OFF" → "ON", the content value of (D0) will become "2" and the content value of (D100) will be moved to (D2), and so forth.
- (D0) records the position where it is written to the FIFO data stack. When the content value of (D0)≥(n-1), the instruction, if enabled again, will not allow data to be written any more, the value of (D0) will remain invariable and the carry flag M9022= "ON".
- This instruction (SFWR) is usually used jointly with the SFRD instruction, specified in the next page, to achieve the write/read control of the FIFO data stack.

FNC 39 SFRD	P	SFRDP (S) (D) (n)	Shift Register Read (FIFO REad)	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S									○	○	○					○
D						○	○	○	○	○	○	○		○		○
n															○	
• n=2 ~ 512																

S : source digit number of the FIFO data stack
D : the device to be read from a FIFO data stack
n : Length of the FIFO data stack

- The data stack of (n) words, headed with (S), is defined as the FIFO data stack. The first device of the FIFO data stack is designated as the indicator. When the instruction is enabled, move the content value of the second device to the device designated by (D), and then, all of the FIFO data stack will be moved a word place to the right, and subtract "1" from the indicator's content value.



- Suppose (D0)=5. When X21= "OFF" → "ON",the content value of (D1) will be moved to (D100), D1 ~ D9 will be moved one word place to the right and the content value of (D0) will become 4, after subtracted by "1". If, again, X21= "OFF" → "ON", the content value of (D1) will be moved to (D101), D1 ~ D9 will be moved one word place to the right and the content value of (D0) will become 3, after subtracted by "1", and so forth.
- When the content value of (D0) equals to "0", the instruction, if enabled again, will not allow read data to be processed any more, the carry flag M9022= "ON" and the value of (D101) will remain invariable.
- This instruction (SFRD) is usually used jointly with the SFWR instruction, specified in the previous page, to achieve the write/read control of the FIFO data stack.

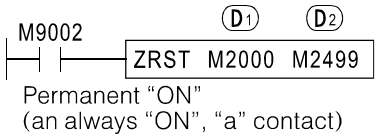
6-6 Data Operation Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
40		ZRST	P	Zone Reset	○	○	○
41		DECO	P	Decode	○	○	○
42		ENCO	P	Encode	○	○	○
43	D	SUN	P	The Sum of active bits	○	○	
44	D	BON	P	Check specified bit status	○	○	
45	D	MEAN	P	Mean	○	○	
46		ANS		Timed Annunciator set	○	○	
47		ANR	P	Annunciator Reset	○	○	
48	D	SQR	P	Square Root	○	○	
49	D	FLT	P	BIN integer → Binary floating point format	○	○	

FNC 40 ZRST	P	ZRSTP (D1) (D2)	Zone Reset	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
D1		○	○	○					○	○	○					○
D2		○	○	○					○	○	○					○

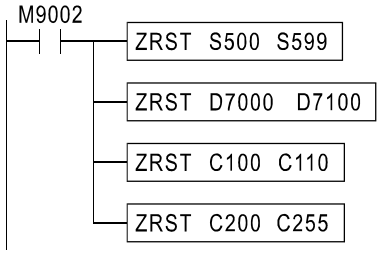
- The ID number of device D1 must be less than or equal to (≤) the device D2.
- D1 and D2 have to designate the device of the same type.



D1: the device starting the Range Reset.

D2: the device terminating the Range Reset.

- When the PLC is under "STOP" → "RUN", M9022 will be "ON" for a Scan Time; All status of coils (M2000 ~ M2499) will be reset to "OFF".



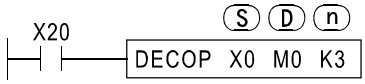
- The devices to be reset by the ZRST instruction consist of various bit devices and word devices.

- (D1) and (D2) have to designate the device of the same type, and the (D1) device's ID number must be less than or equal to (≤) the (D2) device's ID number. Only the device designated by (D2) will be reset if the (D1)'s ID number is greater than the (D2)'s.
- This instruction can reset a 32-bit counter. It's prohibited that (D1) designates a 16-bit counter while (D2) designates a 32-bit counter.

FNC 41 DECO	P	DECOP (S) (D) (n)	Decode	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S	○	○	○	○					○	○	○	○		○	○	○
D		○	○	○					○	○	○	○		○		○
n															○	

- n=1 ~ 8, if d is a bit device.
- n=1 ~ 4, if D is a word device.

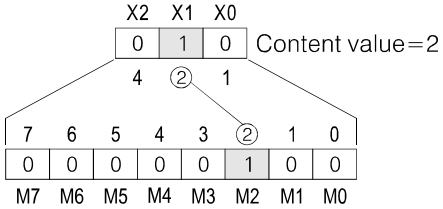


S : Decode source device

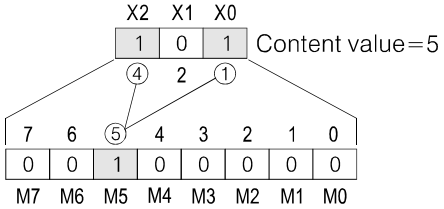
D : Destination device where decode results are stored

n : Length of decode bits

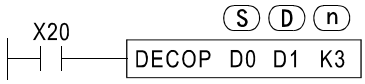
- In this example, (D) designates bit devices, which will occupy 2^n consecutive bit of devices headed with (D) to store decode results.
- Suppose the content value of X0 ~ X2 equals 2. When X20= "OFF" → "ON", the instruction DECO will decode the content value of X0 ~ X2 and move the results to M0 ~ M7, where M2= "ON".



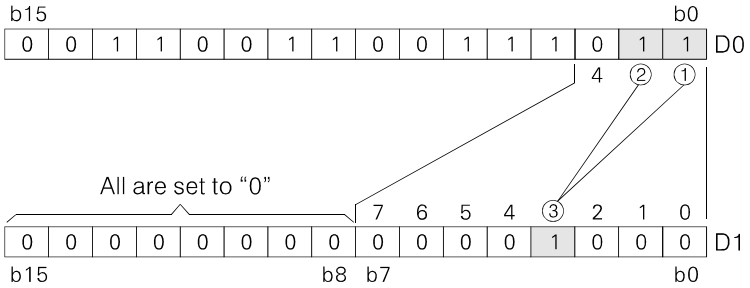
- Suppose the content value of X0 ~ X2 equals 5. When X20= "OFF" → "ON", the instruction DECO will decode the content value of X0 ~ X2 and move the results to M0 ~ M7, where M5= "ON".



- In this example, (D) is a bit device, therefore n=1 ~ 8. When n=8, (D) will occupy 256 bit devices.



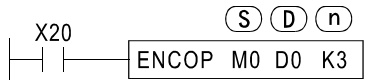
- In this example, (D) designates a bit device, therefore the range of (n) = 1 ~ 4.
- When X20= "OFF" → "ON", the instruction DECO will decode the content value of (b0 ~ b2) of D0 and move the results to (b0 ~ b7) of D1. All unused data bits (b8 ~ b15) will be set to "0".



FNC 42 ENCO	P	ENCOP (S) (D) (n)	Encode	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S	○	○	○	○					○	○	○	○		○		○
D									○	○	○	○		○		○
n															○	

- n=1 ~ 8, if S is a bit device.
- n=1 ~ 4, if S is a word device.

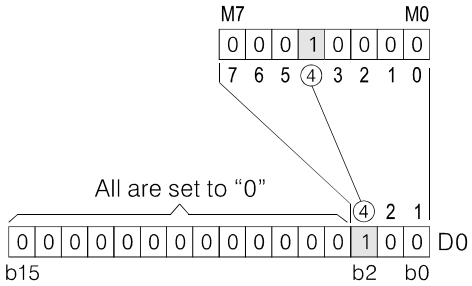


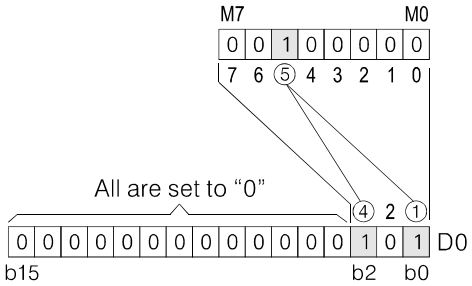
S : Encode source device

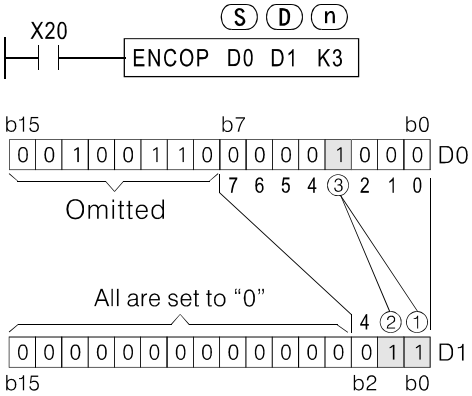
D : Destination device where encode results are stored

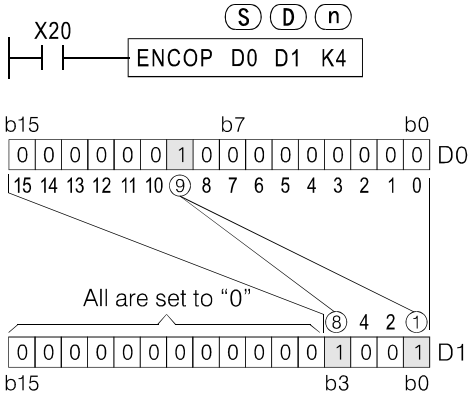
n : Length of encode bits

- When X20= "OFF" → "ON", the instruction ENCO will encode the contents of M0 ~ M7 and move the results to (b0 ~ b2) of D0. All unused data bits (b3 ~ b15) of D0 will be set to "0".
- In this example,(S)is a bit devices,therefore n=1 ~ 8. The effective range of(S)covers 256 bit devices when n=8.







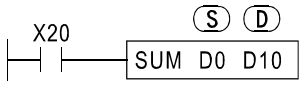


Note:

- If there are more than a bit of the content of(S)equaling "1", the encode will be conducted on the basis of the largest number.
- PLC will identify an operation error if the content of(S)equals "0".
- When the conditional contact turns "OFF", the encode results (status of(D)) will remain.

D	FNC 43 SUM	P	DSUMP (S) (D)	The sum of active ("ON") bits	M	VB	VH
					○	○	

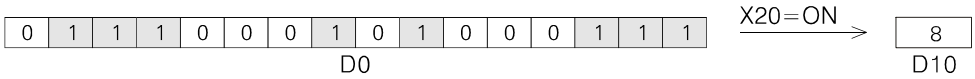
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○



S : Source device

D : Destination device where data are stored

- When X20= "ON", the number of "1" (active) status within the 16 bits D0 are counted, and the among will be stored in D10. If all of the 16 bits of D0 equal "0", then the zero flag M9020= "ON".

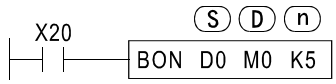


- When a 32-bit instruction DSUM is used, (D) will still occupy 2 registers.

D	FNC 44 BON	P	DBONP (S) (D) (n)	Check specified bit active ("ON") status	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D		○	○	○												○
n															○	

- n=0 ~ 15, for a 16-bit instruction.
- n=0 ~ 31, for a 32-bit instruction.

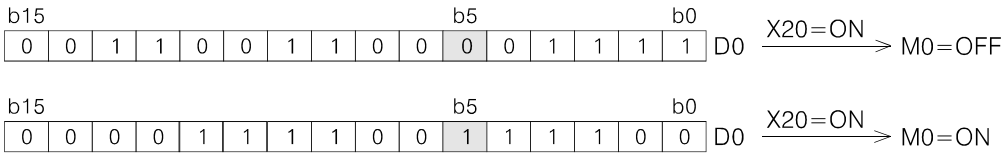


S : Source device.

D : Destination device where specified results are stored.

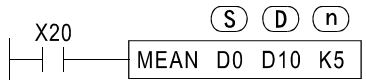
n : the designated position bit to be specified.

- Copy the status of the (n) th bit of the designated source device (S) to the destination device (D).
- When X20= "ON", b5 of D0 will be copied to M0.
- When X20= "OFF", the status of M0 will remain.



D	FNC 45 MEAN	P	DMEANP (S) (D) (n)	Mean	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D						○	○	○	○	○	○	○		○		○
n															○	
• n = 1 ~ 64																



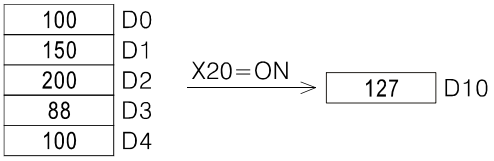
S : Head ID number of source devices to be generated a mean.

D : Destination device where the mean is stored.

n : Number of consecutive devices to be generated a mean.

- To sum up the content values of (n) consecutive devices which headed with (S), then generate a mean value and store it in a device designated by (D).
- When X20= "ON", generate a mean of the content values of consecutive 5 registers (D0 ~ D4) and store it in D10.

$$\frac{(D0) + (D1) + (D2) + (D3) + (D4)}{5} \xrightarrow{X20=ON} (D10)$$



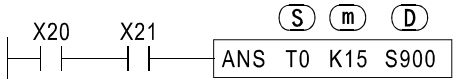
The remainder of the calculated mean is ignored. (Remainder=3)

- Ignore the remainder, if any remainder comes out during the operation process.
- If the designated device's ID number exceeds the range, the device will only be processed within to prescribed range.

FNC 46 ANS		ANS (S) (m) (D)	Timed Annunciator Set	M	VB	VH
				○	○	
FNC 47 ANR	P	ANRP	Annunciator Reset	M	VB	VH
				○	○	

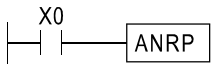
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S									○							○
m															○	
D				○												○

● S=T0～T199 ● m=1～32767 ● D=S900～S999



S : Detect alarm timer
m: Timer configuration
D : Annunciator

- The instruction ANS is used exclusively to drive the instruction of annunciator outputs.
- When X20 and X21 turn “ON” for more than 1.5 seconds simultaneously, the annunciator S900= “ON” (to be driven). After S900= “ON”, X20 or X21 turns “OFF”, the contact of T0 becomes “OFF” and the current value is returned as “0” , but S900 will remain “ON”.
- When both X20 and X21 turn “ON” simultaneously but less than 1.5 seconds, then either one of them turns “OFF”, the current value of T0 will be returned as “0”.
- Do not use a timer which has been assigned to this instruction.



- The instruction ANR is used exclusively to reset the instruction of annunciator. When each time the ANR instruction is operated, annunciators which have been activated are sequentially reset one-by-one.
- When X0= “OFF” → “ON”, the instruction ANR will be executed and the active annunciator will be reset to “OFF”.
- If the instruction ANR is executed and if there are more than one active annunciator, the smallest active annunciator ID number will be reset. When the instruction ANR is executed once again, in this moment the smallest (which was the second smallest) active annunciator ID number will be reset. And so forth to reset other active annunciators.

Application Examples of Timed Annunciator Set

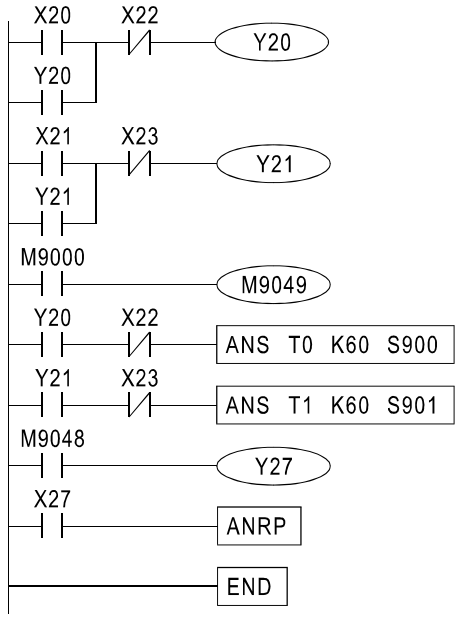
- When the special auxiliary coil M9049= “ON” and any assigned annunciator of S900 ~ S999 is activated, then M9048= “ON” and D9049 will display the annunciator number. If there are more than one annunciator being activated simultaneously, D9049 will display the smallest active annunciator ID number.

- The following chart is a Timed Annunciator Set loop

X20: Forward Switch
X21: Backward Switch
X22: Front End Position Switch
X23: Back End Position Switch
X27: Annunciator Reset Button

Y20: Forward Device
Y21: Backward Device
Y27: Alarm Indicator

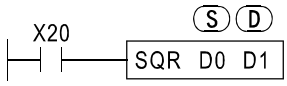
S900: Forward Annunciator
S901: Backward Annunciator



- When the forward switch contact X20= “ON”, the forward device Y20= “ON” and remain. Y20 will become “OFF” when the object reaches the front end position (X22= “ON”).
- When the backward switch contact X21= “ON”, the backward device Y21= “ON” and remain. Y21 will become “OFF” when the object reaches the back end position (X23= “ON”).
- When M9049= “ON”, the alarm monitor will be activated, where M9048 and D9049 are effective.
- If Y20= “ON” for more than 6 seconds and does not reach the front end position, then S900= “ON”.
- If Y21= “ON” for more than 6 seconds and does not reach the back end position, then S901= “ON”.
- The alarm indicator Y27= “ON” if any one of annunciator is “ON”.
- After the problem is cleared, press once X27 to reset the annunciator.

D	FNC 48 SQR	P	DSQRP (S) (D)	Square Root	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S											○				○	○
D											○					○



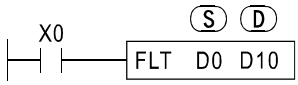
S : Source Device for performing mathematical square roots.

D : Destination device where the result is stored.

- This instruction performs a square root operation on the content value of device (S) and stores the result to the destination device (D).
- We perform a square root operation on the content value of D0 and stores the result at D1 when X20= "ON".
- In the result, only the integer part will remain, while the decimal part will be ignored; If any decimal is ignored, then M9021= "ON".
- Zero Flag M9020= "ON" when the operation result is equal to "0".
- (S) must be a positive; a negative will be determined an error operation by PLC and M9067 will be set "ON".

D	FNC 49 FLT	P	DFLTP (S) (D)	BIN integer → Binary floating point format	M	VB	VH
					○	○	

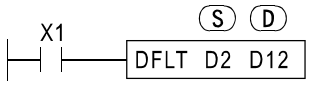
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D											○					○



S : Source data

D : Destination device to store the equivalent float format value

- When X0= "ON",performs the convert operation from the content value of 16 bits register D0 (which is a BIN integer) to a binary floating point number, and copies the converted result to the destination devices (D11,D10).

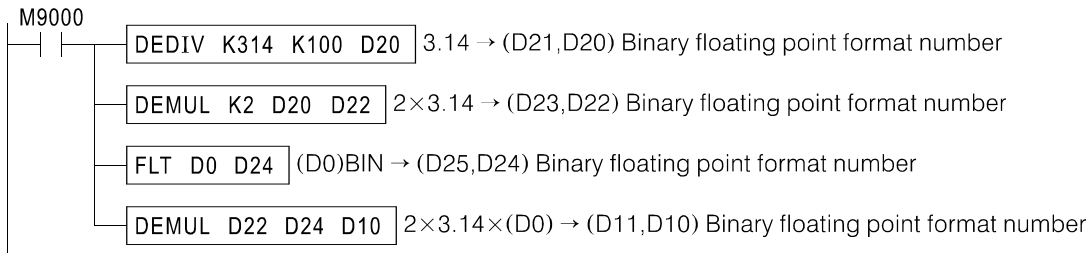


- When X1= "ON", performs the convert operation from the content value of 32 bits registers (D2,D3) (which is a BIN integer) to a binary floating point number, and copies the converted result to the destination devices (D13,D12).

- It is not necessary to use this instruction for constant K or H at floating calculation, because the constant will convert to binary floating point format automatically when the floating calculation is operation.
- A floating point number will occupy two consecutive registers, the format of a floating point number storage in registers, please refer to Section 2-12 "Numerical System".
- Floating point calculation example:
  Use a PLC and FLT instruction to do calculate

$2 \times 3.14 \times$(D0) ⟶ (D11,D10)

BIN integer — Binary floating point format number



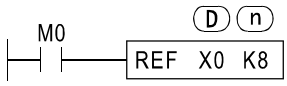
MEMO

6-7 High Speed Processing Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
50		REF	P	I/O Refresh	○	○	○
51		REFF	P	I/O Refresh and Filter Adjust	○	○	
52		MTR		Input Matrix	○	○	
53	D	HSCS		High Speed Counter Set	○	○	○
54	D	HSCR		High Speed Counter Reset	○	○	○
55	D	HSZ		High Speed Counter Zone compare	○	○	
56		SPD		Speed Detection	○	○	○
57	D	PLSY		Pulse Y output	○	○	○
58		PWM		Pulse Width Modulation	○	○	○
59	D	PLSR		Pulse ramp		○	○

FNC 50 REF	P	REFP (D) (n)	I/O Refresh	M	VB	VH
				○	○	○

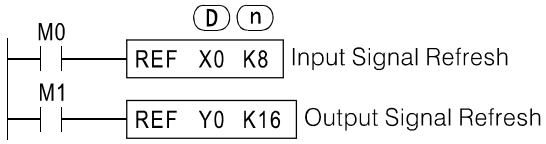
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D	○	○														
n															○	
• D should always designate the device with its last digit of "0" (zero), e.g. X20, X30, Y20, Y30, etc.																
• For M series, n=8 ~ 512 and n should always be a multiple of "8".																
• For VB2 series, n=8 ~ 256 and n should always be a multiple of "8".																
• For VB1 series, n=8 ~ 128 and n should always be a multiple of "8".																
• For VB0 series, n=8 ~ 64 and n should always be a multiple of "8".																
• For VH series, n=8 or 16.																



D : The head address of I/O refresh device

n : The number of I/O refresh devices

- Before PLC performs STEP0 instructions, CPU will read "ON"/ "OFF" status of all input ends once and store them in the data memory. Until the END instruction is executed, all "ON"/ "OFF" status of output signals will be sent to output ends to drive external loadings. This instruction is necessary when we desire to read "ON"/ "OFF" status of the input (X) during the execution of the program or deliver the operation result to the output (Y) immediate.



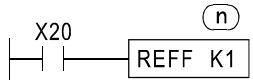
- When M0="ON", the input signal status of X0 ~ X7 will be reloaded to PLC's status data memory. PLC can immediately read the status of X0 ~ X7 while performing this instruction, but the delay (approximately 10ms) on the input contact still remains.
- When M1="ON", the output signal status of Y0 ~ Y17 will be resent to output end contacts from PLC's status data memory. PLC can immediately send the status of Y0 ~ Y17 while performing this instruction, but the delay (by the relay, approximately 10ms) on the output contact still remains.
- (D) should always designates its last digit as "0" (zero), ex. X0, X10, X20, Y0, Y10, etc. (n) Should always be a multiple of "8". Any default value exceeding this range will be regarded as an error.

Series	The range of X for (D)	The range of Y for (D)	The range of (n)
M	X0 ~ X777, total 512 points	Y0 ~ Y777, total 512 points	n=8 ~ 512
VB2	X0 ~ X377, total 256 points	Y0 ~ Y377, total 256 points	n=8 ~ 256
VB1	X0 ~ X177, total 128 points	Y0 ~ Y177, total 128 points	n=8 ~ 128
VB0	X0 ~ X77, total 64 points	Y0 ~ Y77, total 64 points	n=8 ~ 64
VH	X0 ~ X17, total 16 points	Y0 ~ Y17, total 16 points	n=8 or 16

- Use the REF instruction in interrupt subroutines frequently to acquire real-time input/output status.

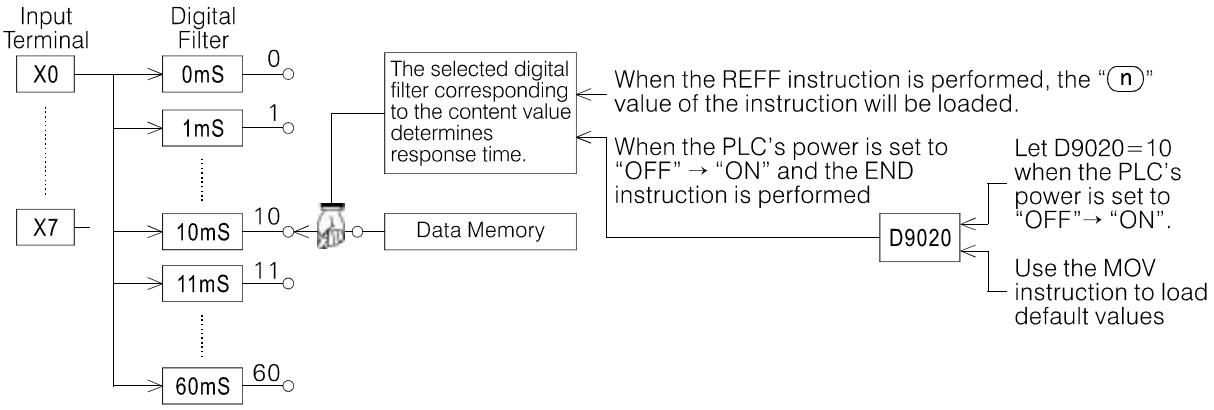
FNC 51 REFF	P	REFFP (n)	I/O Refresh and Filter Adjust	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
n															○	
• n=0 ~ 60																

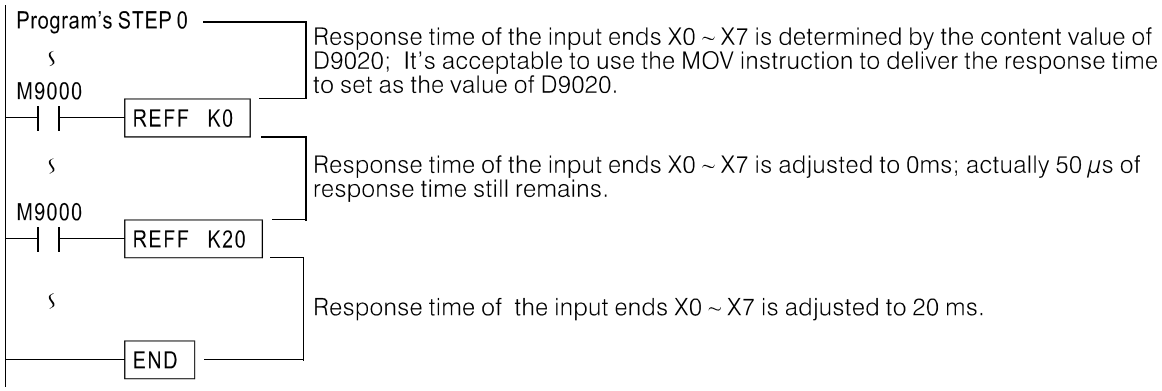


n : the setting for response time (unit = ms)

- When X20="ON", response time for external input end X0 ~ X7 will be changed into 1ms and the "ON" / "OFF" status of X0 ~ X7 will be reloaded into data memory.
- To avoid noise intervention, there will always be a filter with response time approximately 10ms on the PLC's input end to filter out noise; Therefore, if to capture a input signal which with its pulse width less than 10ms, then it will be failed.
- Input contacts of X0 ~ X7 have been equipped with filters on which we can use the REFF instruction to adjust response time. The following figure shows the input configuration of X0 ~ X7:



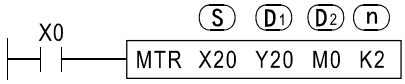
- As shown in the figure above, the input terminals X0 ~ X7 have built-in digital filters with 0 ~ 60 ms. The rules determining response time of the input contacts X0 ~ X7 are described as follows:
  ① When the PLC's power is set to "OFF"→ "ON", the content value of D9020 will be set to 10 and response time will be set to 10ms.
  ② It's acceptable to use the MOV instruction to load the default value to D9020 and to adjust response time.
  ③ Use the REFF instruction to adjust response time during the program execution.

- 

- When the interrupt function, the high speed counter or the SPD (FNC56) instruction is used in the program, response time of the corresponding input terminal will automatically adjusted to 50 μs.

FNC 52 MTR	MTR (S) (D1) (D2) (n)	Input Matrix	M	VB	VH
			○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S	○															
D1		○														
D2	○	○	○													
n															○	
• S should always designate an X with its last digit of "0" (occupies consecutive 8 points).																
• D1 should always designate a Y with its last digit of "0".																
• D2 should always designate a Y, M or S with its last digit of "0".																
• n=2 ~ 8																



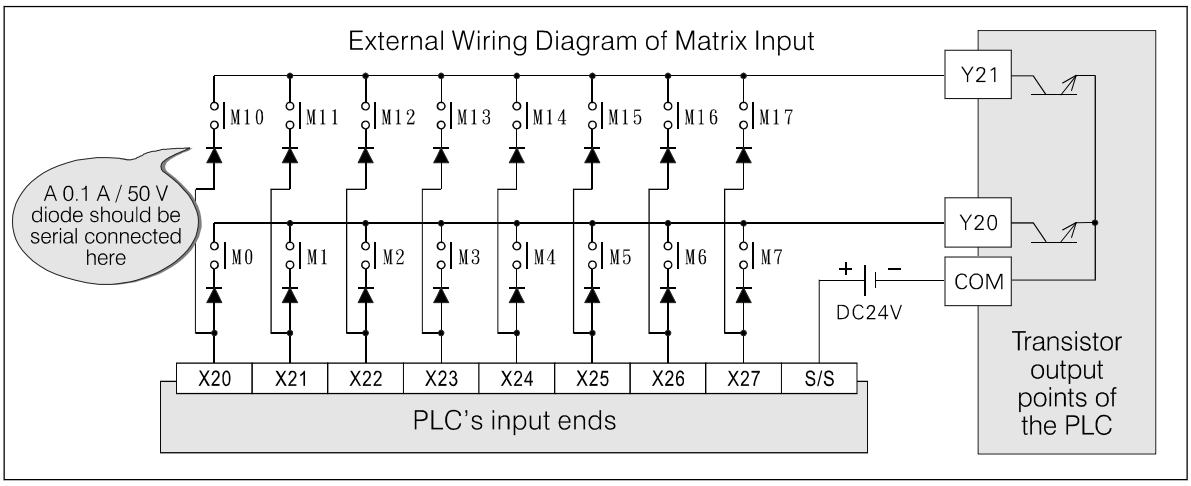
S : the head point for the matrix scan input

D1 : the head point for the matrix scan output

D2 : the head point of the matrix-table (the scan Storage internal coils)

n : number of array rows of the matrix scan

- This instruction reads status through the matrix scan: 8×(n) of external "ON"/ "OFF" status from 8 consecutive input ends which are headed with(S) and(n) output ends are headed with(D1). This matrix scan reads the "ON"/ "OFF" status and reflects on the internal coils headed with(D2).

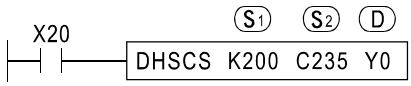


- From the diagram above, X20 ~ X27 and Y20 ~ Y21 constitute two rows array of the matrix input circuit. When X0="ON", the instruction is ready for execution and 16 "ON"/ "OFF" status of (8×2 matrix) array will be read and stored in internal coils of M0 ~ M7 and M10 ~ M17.
- When X0= "OFF", the instruction disables and the status of M0 ~ M7 and M10 ~ M17 remains.
- Using the MTR instruction to read one row of external switches array will takes two scan times. If a scan time is less than 10ms, then reads the status in one row of the array which will takes 20ms to read the status of external "ON"/ "OFF". Maximally, this instruction can connect 8 rows of external switches array. Reading 64 (8×8=64) external switches once will take 16 scan times or 160ms. Therefore, the coordination between external switches response rate and the loading time of the instruction should be considered when this instruction is used.
- The instruction's conditional contacts use M9000 (permanently "ON", "a" contacts) frequently.
- When this instruction performs a scan cycle each time, it will let the Execution Completed Flag M9029= "ON" for one scan time.
- The MTR instruction can be used once during the program.
- This instruction is only recommended for use with transistor output modules.

D	FNC 53 HSCS		DHSCS (S1) (S2) (D)	High Speed Counter Set	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2										○						○
D		○	○	○												○

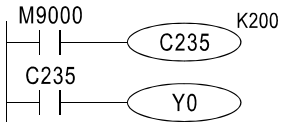
- S2=C235 ~ C255
- D can also designate I0□0,□=1 ~ 6



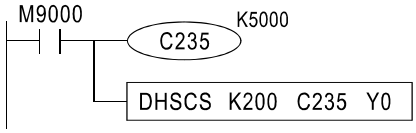
S1 : Compare value
S2 : No. of the selected high speed counter
D : Compare result

- The DHSCS instruction is used to give immediate outputs of High Speed Counter (HSC). HSC receives its high speed counter/pulse of corresponding terminals by using the interrupt input function (for more detailed instruction on HSC, please refer to Section 2-7 "High Speed Counter"). When a HSC is selected by the DHSCS instruction, the current value of HSC(S2) changes (increased/decreased by "1"), the DHSCS instruction will immediately perform the "Compare" operation. When the current value of the HSC is equal to the Compare value (which is selected by(S1)), the device status of(S2) will turn "ON" and then remain the same status ("ON") even if the Compare result becomes unequal. Generally,(D) in this instruction is designated to an output coil Y. When an output coil Y has been designated by(D), the status "ON" will be carry out immediately to the output terminals.
- When this HSCS instruction is used in the VH series PLC and designated(D) to output coils Y, only the output points Y0 ~ Y17 are allowed.
- As the example above, the DHSCS will be enabled when X20="ON". When the current value of C235 changes from 199 to 200 or from 201 to 200, the status of Y0="ON". At the time the status will be sent to output end, and also the status "ON" remains.

Common Output V.S. DHSCS Instruction Output



The timing when the external output end of Y0 is driven, which is affected by the PLC scan time.

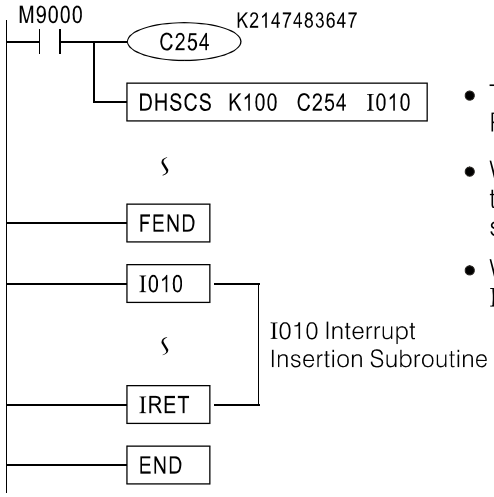


By the function of interrupt, the status of Y0 is immediately output to external output end, irrespective of the PLC scan time. Please notice that: There is output delay of the relay/transistor at the output end.

Note:

- This instruction is a 32-bit instruction; DHSCS should be always entered when the instruction is input.
- There's no limitation on the times used of these instructions DHSCS, DHSCR and DHSZ; However, the total of these instructions performed at the same time should not exceed "6".
- Both the output contacts of High speed counter and Compare output of DHSCS, DHSCR or DHSZ are performed when there is a counter input. The Compare action shall not be performed if the current value of High speed counter is changed by transferring instructions, because there is no counter input signals. Therefore, no Compare output occurs.

High Speed Counter Interrupt



- The device (D) of DHSCS instruction can also designate the Pointer of High speed counter, I0□0, □=1 ~ 6.
- When the current value of C254=100, CPU will jump to the interrupt Pointer I010 to perform the interrupt subroutine.
- When M9059="ON", the High speed counter interrupt of I010 ~ I060 will be blocked.

D	FNC 54 HSCR		DHSCR (S1) (S2) (D)	High Speed Counter Reset	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2										○						○
D		○	○	○												○
• S_2=C235 ~ C255																
• D also available to designate the same High speed counter ID number as S_2 but only when D=S_2.																

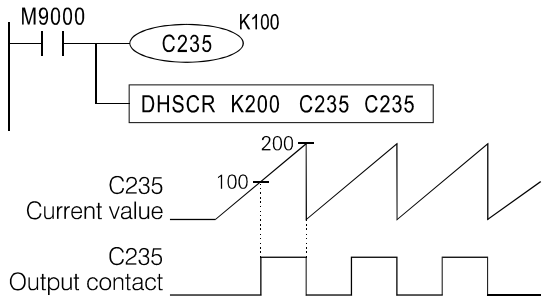
S_1 : Selected compare value

S_2 : Selected high speed counter ID number

D : Use compare result to reset the destination

- The DHSCR instruction is used to give immediate outputs of High speed counter (HSC).
- When this HSCR instruction is used in the VH series PLC and designated (D) to output coils Y, only the output points Y0 ~ Y17 are allowed.
- When X20="ON" and the current value of C235 changes from 199 to 200 or from 201 to 200, the status of Y0="OFF". At the time the status will be sent to output end, and also the status "OFF" remains.

Automatic Reset



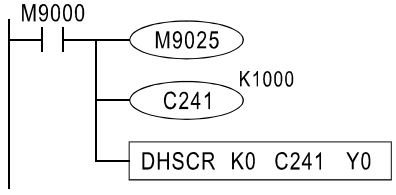
- When (S2) and (D) of the DHSCR instruction designate the same ID number of High speed counter, the counter will perform the self Reset action.
- When the current value of C235=200, C235 will be reset, its current value will be reset to "0", and the output contact will become "OFF".

Note:

- This instruction is a 32-bit instruction; DHSCR should be always entered when the instruction is input.
- There's no limitation on the times used of these instructions DHSCS, DHSCR and DHSZ; However, the total of these instructions performed at the same time should not exceed "6".
- Both the output contacts of High speed counter and Compare output of DHSCS, DHSCR or DHSZ are performed when there is a counter input. The Compare action shall not be performed if the current value of High speed counter is changed by transferring instructions, because there is no counter input signals. Therefore, no Compare output occurs.

About Special Coil M9025

- Some high speed counters have external reset terminals. When the external reset terminal is "ON", the current value of the corresponding high speed counter will be reset to "0" and the output contact will become "OFF". Let M9025="ON", if you desire the reset action to drive external outputs immediately. The following is a sample program.

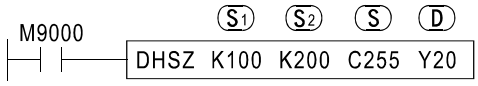


- X1 is the external reset input terminal of C241.
- When X1="ON", the current value of C241 will be reset to "0", the output contact will become "OFF", and the DHSCR instruction will be performed and Y20 will be reset to "OFF".
- If M9025= "OFF" and X1="ON", the current value of C241 will be reset to "0", the output contact will become "OFF", and the DHSCR instruction will not be performed and the status of Y20 will remain the same.

D	FNC 55 HSZ		DHSZ (S1) (S2) (S) (D)	High Speed Counter Zone compare	M	VB	VH
					○	○	

Operation 1: High Speed Counter Current Value Against To a Specified Range

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
S										○						○
D		○	○	○												○
• S=C235 ~ C255																
• D occupies 3 consecutive points, if D is designated to a Y, then D shall be Y□□0 ~ Y□□5																



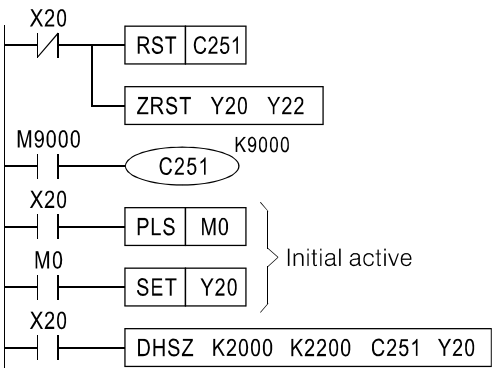
S1 : Lower limit of Zone Compare
S2 : Upper limit of Zone Compare
S : High speed counter ID number
D : Compare Result

- All of the counting value and result outputs of this instruction are processed with interrupt insertion, Y20 ~ Y22 will immediately output irrespective of scan time. Results of Zone comparison are shown as follows:
  When K100>the current value of C255, then Y20="ON".
  When K100≤the current value of C255≤K200, then Y21="ON".
  When K200<the current value of C255, then Y22="ON".
- If (S1)>(S2), then the value of (S1) will become both of the Upper/Lower limits to compare with (S).

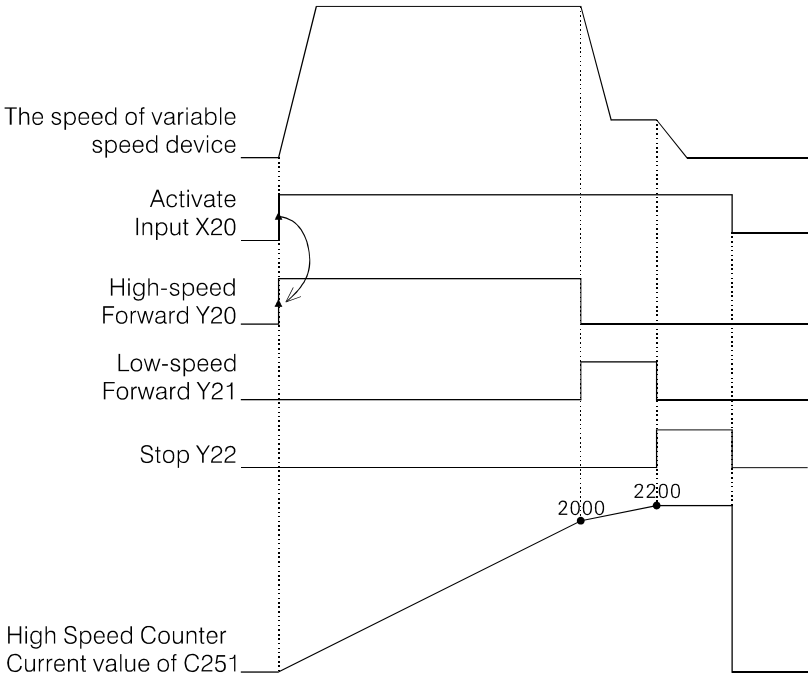
Notes for all modes (operation 1~3):

- This instruction is a 32-bit instruction; DHSZ should be always entered when the instruction is input.
- There's no limitation on the times used of these instructions DHSCS, DHSCR and DHSZ; However, the total of these instructions performed at the same time should not exceed "6".
- Both the output contacts of High speed counter and Compare output of DHSCS, DHSCR or DHSZ are performed when there is a counter input. The Compare action shall not be performed if the current value of High speed counter is changed by transferring instructions, because there is no counter input signals. Therefore, no Compare output occurs.
- If (D) of the DHSZ instruction is designated to a Y, the assigned ID number should be Y□□0 ~Y□□5 , rather than Y□□6 or Y□□7 (e.g. Y20, Y25 are acceptable while Y26, Y27 are not).

Use the HSZ instruction to perform high/low speed stop control



- C251 is an A/B phase high speed counter, X0 is an A-phase pulse input, and X1 is a B-phase pulse input.
- X20 is a signal for activation.
- The DHSZ will have compare outputs only when there is counting pulse entering into C251. So when X20="OFF"→"ON" (the initial active signal of the left-side program), it will activates the motor operation (let Y20= "ON"). At the very beginning, the motor operation will produces a counting pulse and feedback it to the High speed counter. And then, performs the corresponding Compare results of Y20 ~ Y22.

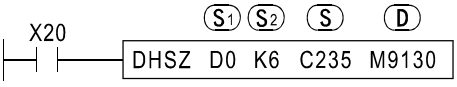


Operation 2: The HSZ Instruction's Multiple Point Compare Mode (When D=M9130)

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	Z	K,H	VZ index
S_1											○					○
S_2															○	○
S										○						○
D			○													

- S_1 occupies 4×S_2 consecutive Registers; S_2=K1 ~ K128; S=C235 ~ C255;

When the DHSZ instruction's (D) is designated to M9130, the instruction will perform Compare outputs between the current value of High speed counter and the setting values of comparison data table. In this mode, devices of each operand are shown as follows:



S_1 : Head device ID number of the Compare table, designates Data Register D only

S_2 : Number of Compare data groups, designates K1 ~ K128 only

S : High speed counter ID No., designates C235 ~ C255 only

D : Mode designation, designates M9130 only

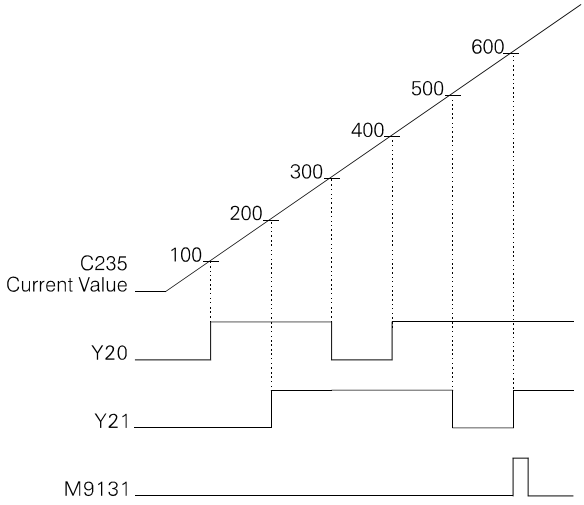
Comparison Data Table

D9130

	Comparison Data		Designate "Y" Output Device	SET/RST Designated "Y" Device	Table of Record Number
	Lower Reiister	Upper Register			
S_1	D0 K100	D1 K0	D2 H20	D3 K1	0
	D4 K200	D5 K0	D6 H21	D7 K1	1
	D8 K300	D9 K0	D10 H20	D11 K0	2
S_2	D12 K400	D13 K0	D14 H20	D15 K1	3
	D16 K500	D17 K0	D18 H21	D19 K0	4
	D20 K600	D21 K0	D22 H21	D23 K1	5

H20 represents Y20
H21 represents Y21

K0: Reset to "OFF"
K1: Set to "ON"



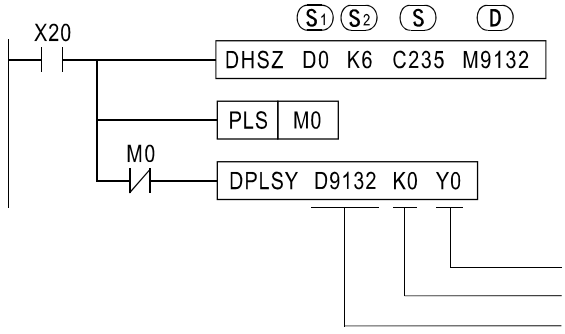
- When X20="ON", the instruction begins to be performed. The Comparison Data Table is processed by one "Record number" at a time. A comparison between the current value of High speed counter C235 (which is designated by) and the content value of Comparison data (D1, D0) in the first group (Record 0) is started. If the comparison is equal, Y20 will be set to "ON" and output immediately. And also, the content value of Record Number D9130 will be increased by "1" (turn into "1").
  Then, the current value of C235 begins to be compared to the content value of Comparison data (D5, D4) of second Group (Record 1). If the comparison is also equal, Y21 will be set to "ON" and output immediately. And also, the content value of Record Number D9130 will be increased by "1" (turn into "2"). Then, the subsequent Compare will be proceeded accordingly, until the data compare of the last group is equal while Execution Completed Flag M9131="ON" for a scan time. Later D9130 will be reset to "0" and the data Comparison of first group will be performed again.
- When X20="ON"→ "OFF", the instruction will be disabled, the content of Record Number D9130 will be cleared as "0", but while the output coilis "ON"/ "OFF" status will remain.
- The instruction's Compare operation and output actions are processed by interrupt function.
- The instruction can only be used once in a program.

Operation 3: The Frequency Control Mode Combining HSZ and PLSY (When D=M9132)

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	Z	K,H	VZ index
S_1											○					○
S_2															○	○
S										○						○
D			○													

- S_1 occupies 4×S_2 consecutive Registers; S_2=K1 ~ K128; S=C235 ~ C255

When the DHSZ instruction's (D) is designated to M9132, and assemble with the DPLSY instruction as follows, which performs the function that using the current value of High speed counter to control the PLSY pulse output frequencies.



S_1 : Head device ID number of the Compare table, designates Data Register D only

S_2 : Number of Compare data groups, designates K1 ~ K128 only

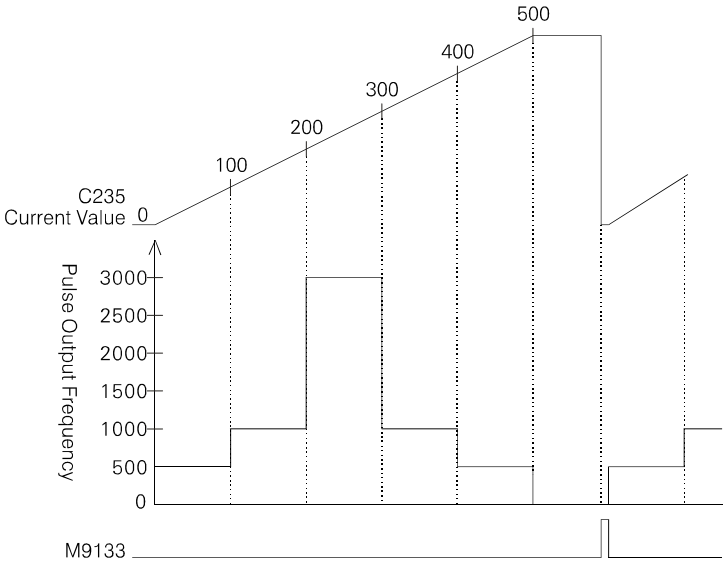
S : High speed counter ID No., Designates C235 ~ C255 only

D : Mode designation, designates M9132 only

Comparison Data Table

D9131

Comparison Data		Pulse Output Frequency 0 ~ 20KHz/ 0 ~ 7KHz	Table of Record Number
Lower Reiister	Upper Register		
D0 K100	D1 K0	(D3,D2) K500	0
D4 K200	D5 K0	(D7,D6) K1000	1
D8 K300	D9 K0	(D11,D10) K3000	2
D12 K400	D13 K0	(D15,D14) K1000	3
D16 K500	D17 K0	(D19,D18) K500	4
D20 K0	D21 K0	(D23,D22) K0	5



- When X20="ON", the instruction begins to be performed. The Comparison Data Table is processed by one "Record number" at a time. In the beginning, the content value of Table D9131= "0". According to the content value (D3, D2) of the Comparison Data Table, Y0 is assigned to output 500Hz pulses.
  Besides, a comparison between the current value of High speed counter C235 (which is designated by (S)) and the content value of Comparison data (D1, D0) in the first group (Record 0) is started. If an equal comparison is given, the content value of Record Number D9131 will be increased by "1" (turn into "1"). And then, Y0 outputs 1000Hz pulses according to (D7, D6) of the Comparison Data Table and the current value of C235 begins to be compared to the content value of Comparison data (D5,D4) in the second Group (Record 1). If the comparison is also equal, the content value of D9131 will be increased by "1" (turn into "2"). Then, the subsequent Compare will be proceeded accordingly,until the data compare of the last group is equal while Execution Completed Flag M9133="ON" for a scan time. Later D9131 will be reset to "0" and the data Comparison of first group will be performed again.

- When X20=“ON”→“OFF”, the instruction will be disabled, the content of Record Number D9131 will be cleared as “0”.
- The instruction can only be used once in a program.
- When this instruction is performed, the PLSY instruction will be not performed until the first scan is finished, and the preparation of the data in the Comparison Data Table must be completed before the first scan to the END instruction.
- D9131: Record Number Counter for the Comparison Data Table

  D9132: In this frequency control mode, it will using the content value of D9131 to select frequency which is the corresponding pulse output frequencies in the Comparison Data Table, and put the selected frequency into (D9133, D9132) registers.

  D9134: In this frequency control mode, it will using the content value of D9131 to select corresponding Comparison Datas in the Comparison Data Table, and put the selected datas into (D9135, D9134) registers.

FNC 56 SPD	SPD (S1) (S2) (D)	Speed Detection	M	VB	VH
			○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1	○															○
S2					○	○	○	○	○	○	○	○		○	○	○
D									○	○	○					○

- S1=X0 ~ X5
- D occupies 3 consecutive points

```
          (S1)  (S2)  (D)
X20 ──┤├──┬──────────────┐
          │ SPD  X0  K1000  D0 │
          └──────────────┘
```

S1 : Exterior pulse input end
S2 : Time frame of receiving pulses (unit:ms)
D : Detection result

- Within the time frame (which is designate by(S2) and unit=ms), calculate the number of pulses coming from the exterior input end (which is designate by(S1)) and store the result in the register (which is designate by(D)).
- When X20="ON", D1 begins to accumulate the number of pulses input from the exterior input end X0. After 1000ms of the time frame, store the accumulated results to D0, then clear the current count value of D1 as "0". And then, once again, re-calculate the number of input pulses from X0.
- D2 displaies the Timer's remaining time (unit=ms).
- The main purpose of the instruction is to get the rotation rate of the rotation facility. The rotation rate can get easily from using the content value of D0:

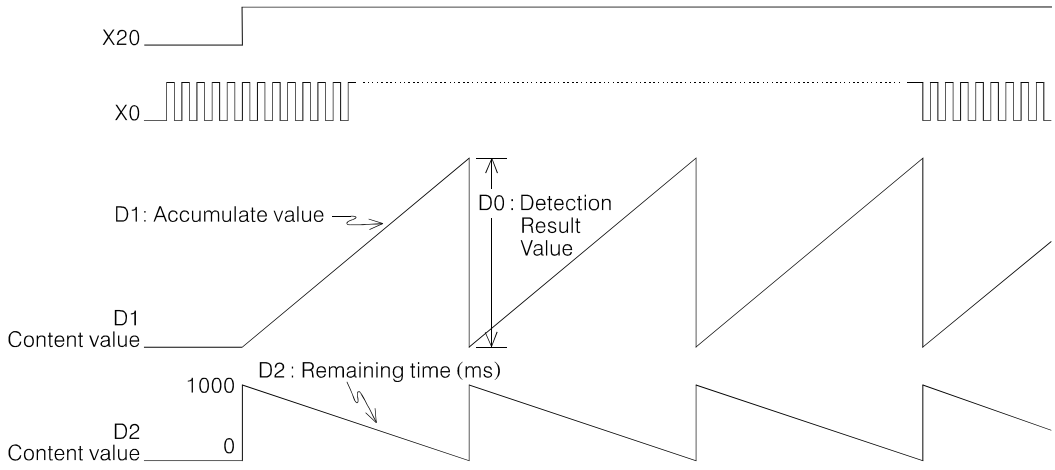
$$N=\frac{60\times(\text{Content value of D0})}{nt}\times 10^3\ (\text{rpm})$$

N : Rotation rate
n : Number of pulses generated from a rotation of the rotation facility
t : Content value designated by(S2)

As in the equation referred above, let n=100, (D0)=3,000, then we will have

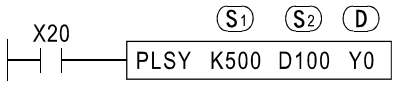
$$N=\frac{60\times 3000}{100\times 1000(\text{ms})}\times 10^3=1800\ (\text{rpm})$$

- The exterior input end designated by the instruction's(S1) cannot be used as the pulse input terminal or the exterior interrupt insertion signal for High speed counter.
- The max. frequencies of input pulses for the instruction's exterior input end X0 ~ X5 will be 10KHz. But, all the SPD instruction's and High speed counter's total counting frequencies should be no faster than 20KHz.



D	FNC 57 PLSY		DPLSY (S1) (S2) (D)	Pulse output	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
D		○														○
M series	● S1=2～20000				● 16-bit instruction S2=0～32767					● 32-bit instruction S2=0～2147483647				● D=Y0 or Y1		
VB series	● S1=2～7000				● 16-bit instruction S2=0～32767					● 32-bit instruction S2=0～2147483647				● D=Y0 or Y1		
VH series	● S1=2～7000				● 16-bit instruction S2=0～32767					● 32-bit instruction S2=0～2147483647				● D=Y0		



S1 : Pulse output frequency
S2 : Number of pulse outputs
D : Pulse output point

- When X20="ON", Y0 outputs the specified quantity (D100's content value) of pulses at the 500Hz frequency rate (500 pulses per second).

  (S1) designates the output pulse frequency range. (M series from 2 to 20,000Hz; VB and VH series from 2 to 7,000Hz)

  (S2) designates the number of output pulses

  For a 16-bit instruction, the specified range will be 1～32,767 pulses.

  A 32-bit instruction, the specified range will be 1～2,147,483,647 pulses.

  If (S2) is set to "0", the quantity of pulses is unlimited for continuous outputs.

  (D) designates the pulse output point (M and VB series can use Y0 or Y1 only; VH series can use Y0 only).

- The signal pulse is described as having a 50% duty cycle (it is "ON" for 50% of the pulse and consequently "OFF" for the remaining 50%). CPU transfers pulses to output ends immediately by the interrupt mode.
- When the quantity of pulse outputs (which designated by (S2)) are completed, then M9029="ON" for a scan time.
- Special Register D9137 (Upper 16 bits), D9136 (Lower 16 bits) will display the total output pulses of the PLSY instruction.

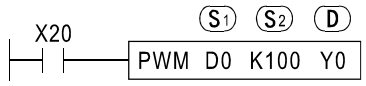
  Special Register D9141 (Upper 16 bits), D9140 (Lower 16 bits) will display the PLSY instruction's output pulses to Y0.

  Special Register D9143 (Upper 16 bits), D9142 (Lower 16 bits) will display the PLSY instruction's output pulses to Y1.
- When the conditional contact X20 becomes "OFF" during the pulse output, pulse outputs will be stopped and the pulse outpoint (Y0 or Y1) will also turns "OFF"; When X20 becomes "ON" again, the pulse generating will be restored from the first pulse.
- During the instruction execution, it's possible for the instruction to change the content value of (S1) through the program; However, changing (S2) will not become effective until the current operation has been completed.
- The instruction can only be used once in a program.

FNC 58 PWM	PWM (S1) (S2) (D)	Pulse Width Modulation	M	VB	VH
			○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
D		○														○

- S1=0 ~ 32767
- S2=1 ~ 32767
- D=Y0 or Y1 (VH series D=Y0)

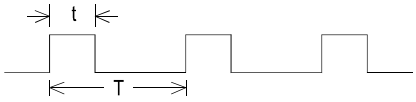


S1 : Output Pulse "ON" width, t=0 ~ 32,767 ms

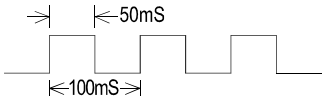
S2 : Output Pulse cycle distance, T=1 ~ 32767ms

D : Pulse output point

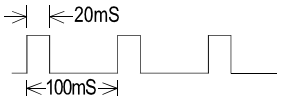
- The PWM instruction is operated as an instruction generating t/T pulse width modulation characteristics of the sequence diagram shown in the right.



- When the conditional contact is "ON", a pulse with a cycle distance of "T" (designated by (S2)) and the "ON" pulse width of "t" (designated by (S1)) will be output at the output point which designated by (D).
- When X20="ON" and suppose D0=50, then Y0 will output the following pulses



- When X20="ON" and suppose D0=20, then Y0 will output the following pulses

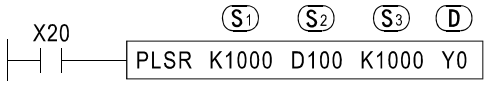


- If X20 becomes "OFF", Y0 will also become "OFF".
- If "t" is larger than "T", an operation error will occur.
- The PWM instruction will be operated only once in the program.
- The pulse output point specified by the instruction cannot overlap the output point which specified by the PLSY or PLSR instruction.

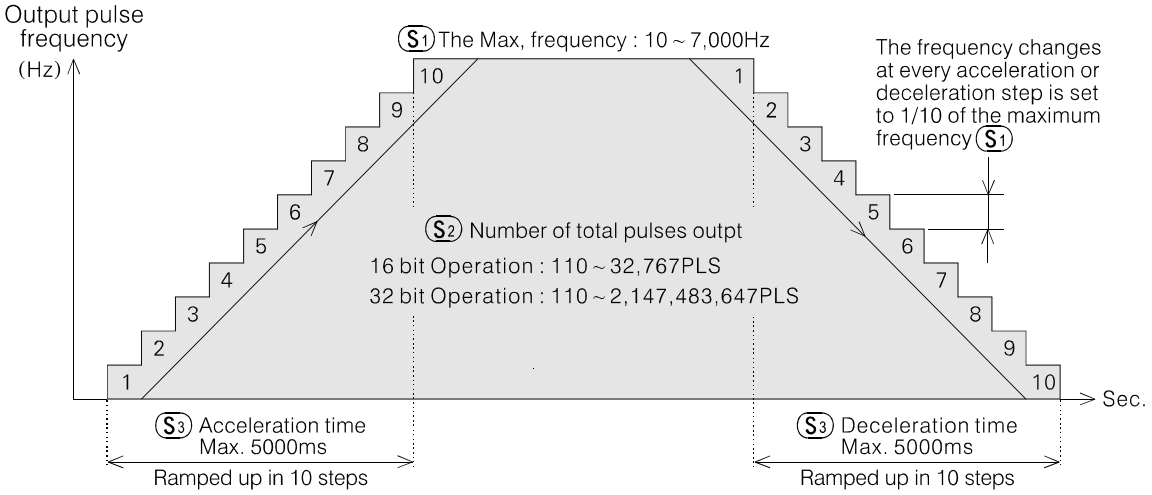
D	FNC 59 PLSR		DPLSR (S1) (S2) (S3) (D)	Pulse ramp	M	VB	VH
						○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
S3					○	○	○	○	○	○	○	○		○	○	○
D		○														○

• S1=10 ~ 7000 • 16-bit instruction S2=110 ~ 32767 • 32-bit instruction S2=110 ~ 2147483647 • D=Y0 or Y1



S1 : Maximun pulse output frequency
S2 : Number of total pulses output
S3 : Ramp time for acceleration or deceleration
D : Pulse output point



- When X20= "ON", Y0 outputs the specified quantity (D100's content value) of pulses, the previous diagram is for showing the development of output frequency.

  (S1) designates the maximun output pulse frequency.
  It may use the frequency range from 10 to 7,000Hz, and also the frequency should be set to a multiple of 10.

  (S2) designates the number of output pulses.
  For a 16-bit instruction, the specified range will be 110 ~ 32,767 pulses;
  For a 32-bit instruction, the specified range will be 110 ~ 2,147,483,647 pulses.

  (S3) designates the ramp time for acceleration or deceleration. (unit=ms)

  The available range is: $\frac{100,000}{(S1)} \leq (S3) \leq 5,000$

  If set the value of S3 is less than $\frac{100,000}{(S1)}$, the error range of the acceleration/deceleration steps' timing become larger.

  And also, please set the value of (S3) more than 10 times of the maximum program scan time (the content value of D9012). If the setting is less than this, then the timing of the acceleration/deceleration steps become uneven.

  (D) designates the pulse output device is limited to Y0 or Y1 only and the output point should be transistor type.

- This instruction may use the range of output frequency is from 10 to 7,000Hz. When the frequencies of the maximum output pulse or the acceleration/deceleration steps are exceeded the range, it will automatically adjust the frequencies to this range.
- When the quantity of pulse outputs (which designated by(S2)) are completed, then M9029＝“ON” for a scan time.
- Special Registers D9137 (Upper 16 bits) and D9136 (Lower 16 bits) will display the total output pulses of the PLSY and PLSR instructions.

  Special Registers D91341 (Upper 16 bits) and D9140 (Lower 16 bits) will display the PLSY and PLSR instructions output pulses to Y0.

  Special Registers D9143 (Upper 16 bits) and D9142 (Lower 16 bits) will display the PLSY and PLSR instructions output pulses to Y1.

  The content value of Special Registers above can use the instruction DMOV K0 D91□□to reset it.
- When the conditional contact X20 becomes ”OFF” during the pulse output, pulse outputs will be stopped and the pulse output point (Y0 or Y1) will also turns ”OFF”; When X20 becomes “ON” again, the pulse generating will be restored from the first pulse.
- During the instruction execution, to change any parameter in this instruction is useless.
- The instruction can only be used once in a program.
- The Y0 and Y1 output points which are driven by PLSY or PLSR instruction can not output pulse at the same time.

MEMO

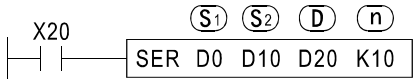
6-8 Handy Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
61	D	SER	P	Search	○	○	
62	D	ABSD		Absolute Drum Sequencer	○	○	○
63		INCD		Incremental Drum Sequencer	○	○	○
64		TTMR		Teaching Timer	○	○	
65		STMR		Special Timer	○	○	
66		ALT	P	Alternate state	○	○	○
67		RAMP		Ramp variable value	○	○	○
69		SORT		Sort tabulated data	○	○	

D	FNC 61 SER	P	DSERP (S1) (S2) (D) (n)	Search	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○					○
S2					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○					○
n											○				○	

• For a 16-bit instruction, n=1 ~ 256 • For a 32-bit instruction, n=1 ~ 128 • D occupies 5 consecutive devices



S1 : Head device ID number of a defined data stack to be searched

S2 : Parameter data to be searched

D : Searched result's storage head device ID number

n : The stack length of the searched data

- The data stack is assigned by "(n)" consecutive devices which headed with(S1). Compare the content value of the device specified by(S2)to each device in the data stack, and store the comparison result into 5 consecutive devices headed with(D).
- For a search data stack formed by D0 ~ D9. When X20= "ON", compare D10 with D0~D9 and store the result into D20 ~ D24. (Assune the content value of parameter D10=100.)

The result of the search

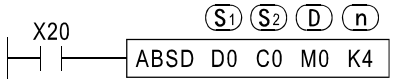
	Data position number	Data stack for searching	Content value of D0 ~ D9	Compared Data	Comparison Data
	0	(S1) D0	100		Equal value
	1	D1	120		
	2	D2	100		Equal Value
	3	D3	85	(S2)	
(n)	4	D4	125	D10	
	5	D5	60	100	Min. Value
	6	D6	100		Equal Value
	7	D7	95		
	8	D8	100		Equal Value
	9	D9	210		Max. Value

Result Storage Device	Content Value	Description
(D) D20	4	Total number of the equal comparison result
D21	0	Data position number of the first equal value
D22	8	Data position number of the last equal value
D23	5	The Min. value data position number
D24	9	The Max. value data position number

- (D) will record the larger data position number when there's more than one minimum or maximum value in the data stack.
- All the content values of D20 ~ D22 will be "0" when there's no equal value.
- For a 32-bit instruction,(S1), S2 and(D) will designate a 32-bit register while(n) will designate a 16-bit register.

D	FNC 62 ABSD		DABSD (S1) (S2) (D) (n)	Absolute Drum Sequencer	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○					○
S2										○						○
D		○	○	○												○
n															○	
• When S1 designates K*n*X, K*n*Y, K*n*M and K*n*S, where n of K*n* should be "4" for a 16-bit instruction and should be "8" for a 32-bit instruction; the ID number of X, Y, M and S should be a multiple of "8"																
• A 16-bit instruction S2 = C0 ~ C199, a 32-bit instruction S2 = C200 ~ C255 • n = 1 ~ 64																



S1 : Head device ID number of the comparison table
S2 : The ID number of the counter
D : Head device ID number of the comparison result
n : Number of comparison section groups

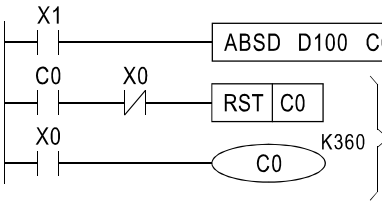
- The instruction is a Multi-Section Compare instruction and generally is operated for multi-section absolute drum sequencer.

	Lower Limit	Upper Limit	Comparison Value	Comparison Result
	(S1) D0 = 50	D1 = 200		(D) M0 = 1
	D2 = 0	D3 = 50	(S2)	M1 = 0
(n)	D4 = 80	D5 = 120	C0 = 100	M2 = 1
	D6 = 120	D7 = 300		M3 = 0

- When X20 = "ON", the current value of the selected counter C0 is compared against a user defined data table [(D0, D1), (D2, D3), (D4, D5) and (D6, D7) 4 groups of upper/lower limit], and the results are stored on M0 ~ M3 respectively.
  If [Lower Limit ≤ Comparison Value ≤ Upper Limit], the corresponding output point will be turned "ON"; Otherwise, the comparison value is not placed between Upper Limit and Lower Limit, the corresponding output point will be turned "OFF".
- When X20 = "OFF", the status, "ON"/ "OFF", of M0 ~ M3 will remain.

A Program Example

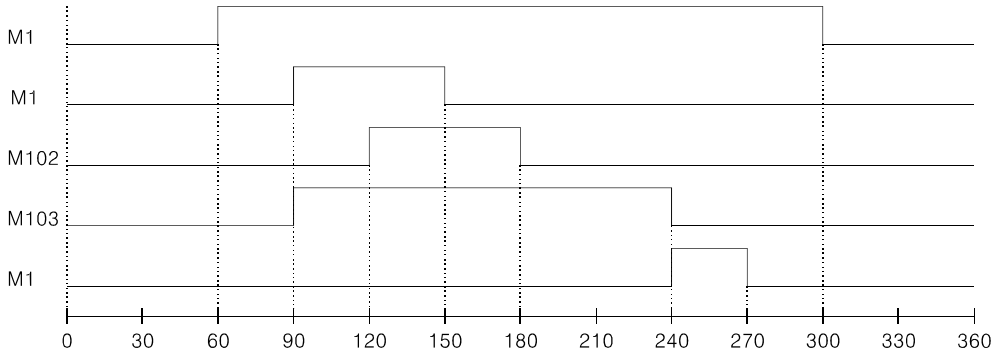
Suppose that a drum-controlled rotor sends a pulse to the input terminal X0 when it rotates by one degree, then the following program will perform the checkup and control actions of the drum degree.



When X1 = "ON", activate Multi-Section Compare

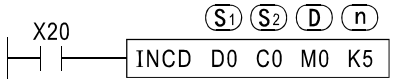
This section of program enables C0 to be the counter of rotor which is depending on counting the rotation degree of the rotor.

Lower Limit	Upper Limit	Comparison Value	Comparison Result
D100=60	D101=300	C0	M100
D102=90	D103=150		M101
D104=120	D105=180		M102
D106=90	D107=240		M103
D108=240	D109=270		M104



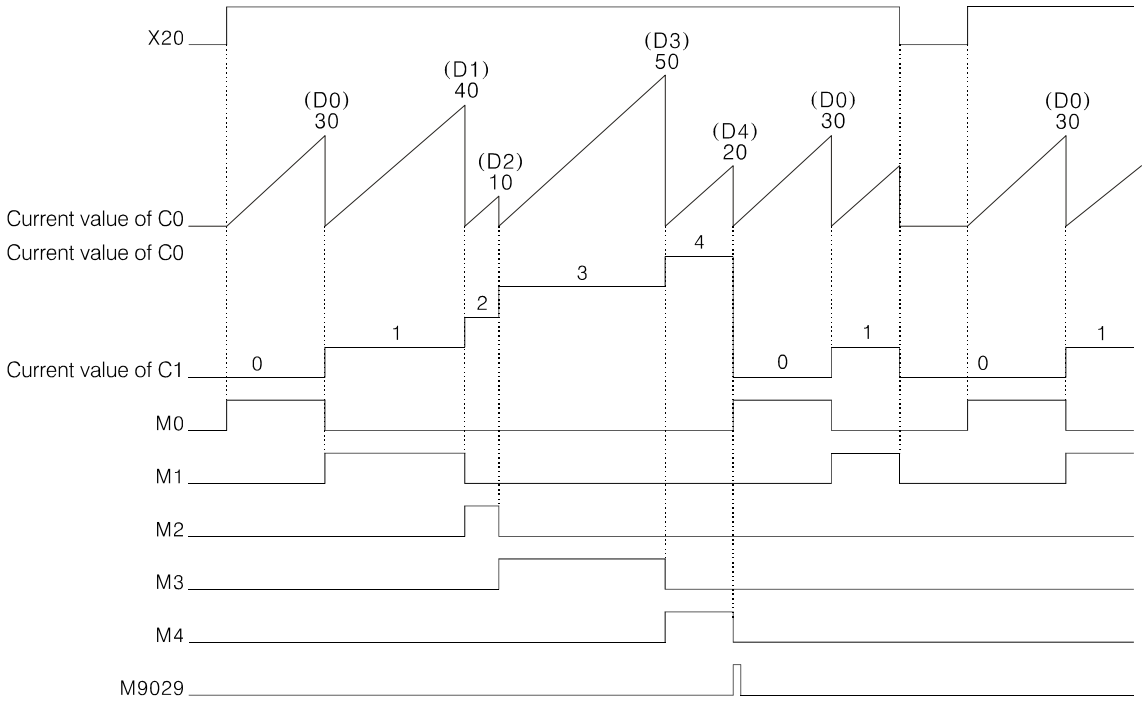
FNC 63 INCD	INCD (S1) (S2) (D) (n)	Incremental Drum Sequencer	M	VB	VH
			○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○					○
S2										○						○
D		○	○	○												○
n															○	
• When S1 designates K*n*X, K*n*Y, K*n*M and K*n*S, where n of K*n* should be "4" and the number of X, Y, M and S should be a multiple of "8"																
• S2 = C0 ~ C198 • n = 1 ~ 64																



S1 : Head device ID number of the comparison table
S2 : ID number of the counter
D : Head device ID number of the comparison result
n : Number of comparison section groups

- The INCD instruction is a multi-section incremental drum-controlled instruction.
- If (S1) designates D0, n=5 and (D) designates M0, then the comparison values are stored in D0 ~ D4, while M0 ~ M4 serve as the outputs. Suppose that (D0)=30, (D1)=40, (D2)=10 and (D4)=20.
- For detailed actions of the INCD instruction, please refer to the following sequence diagram.

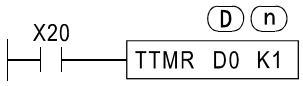


- (S2) will occupies two consecutive ID number counters.
- For a multi-section incremental comparison output, Execution Completed Flag M9029 will turn "ON" for a scan time while a circulation is completed.
- When X20= "ON" → "OFF", the current values of C0 and C1 will be reset to i0i and M0 ~ M4 will be turned "OFF".

FNC 64 TTMR	TTMR (D) (n)	Teaching Timer	M	VB	VH
			○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D											○					○
n															○	

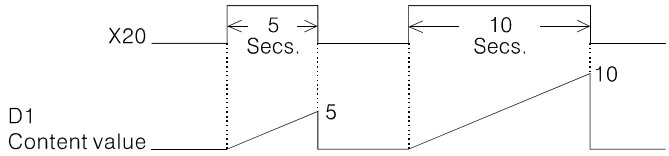
- n=0 ~ 2
- D occupies 2 consecutive registers



D : The ID number of the register which can store the timed data of "ON" duration (ex. From a push button switch)

n : Setting for multiplier

- Suppose X20 is the external push button switch.
- When X20= "ON" (is pressed), the content value of D1 will respond (in secs) to the duration of X20= "ON".
  If X20="ON" for 5 seconds, then D1=5. Because n=1, so D0=50.
  If X20="ON" for 10 seconds, then D1=5. Because n=1, so D0=100.



- The content value of D0 is determined by the content value of D1 and "n"; their correlation is:

(n)	Content value of D1	Content value of D0	
0	10 (Supposed Value)	10×1=10	When n=0, (D0)=(D1)×1
1		10×10=100	When n=1, (D0)=(D1)×10
2		10×100=1000	When n=2, (D0)=(D1)×100

Accordingly, it is easily to use the content value of D0 become the setting value of a Timer T via a proper setting value of "(n)".

n=1 can be applied to a 100 ms unit Timer

n=2 can be applied to a 10 ms unit Timer

- When X20= "ON" → "OFF", the current value of D1 will be reset to i0i but the content value of D0 will remain.

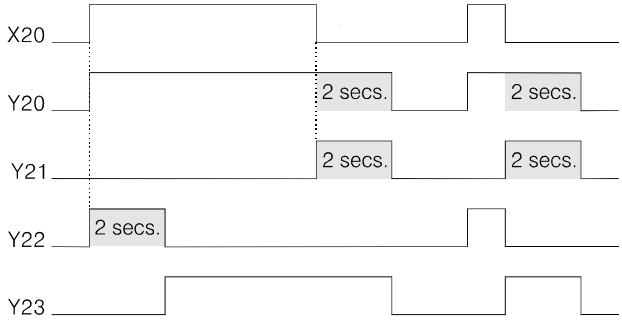
FNC 65 STMR	STMR (S) (m) (D)	Special Timer	M	VB	VH
			○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S									○							○
m															○	
D			○	○	○											○

- S=T0～T199
- m= 1～32767
- D occupies 4 consecutive devices

S : ID number of designated Timer
m : Setting value of the Timer (unit=100ms)
D : Head ID number of the output device

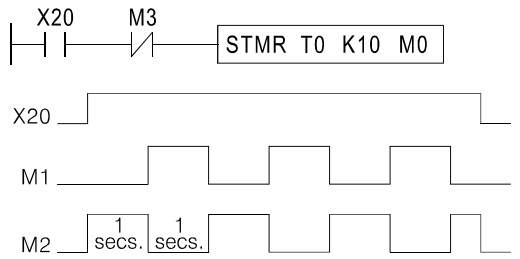
- The STMR instruction is operated exclusively to produce an Off-delay, a trigger and a flashing circuit.
- When X20= "ON", the STMR instruction starts to be performed. As (m)=20, the T0 become a 2 seconds setting value Timer.



- Y20 is an Off-delay output.
- When Y21 is an input signal turned from "ON" to "OFF", a trigger for one shot timer output will be enabled.
- Y22 and Y23 are designed for output signals exclusively composing the flashing circuit. The following example is a practical approach for the flashing circuit.

- In the program, do not reuse the Timer ID number which has been used by this instruction before.

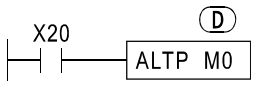
Flashing Circuit



- Perform a serial link "b" Contact of M3 after X20, then M1 and M2 will perform the flashing circuit.

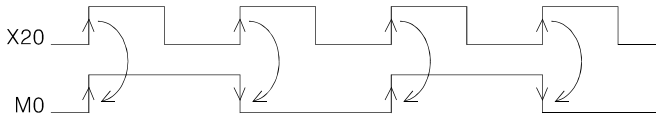
FNC 66 ALT	P	ALTP (D)	Alternate State	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D		○	○	○												○

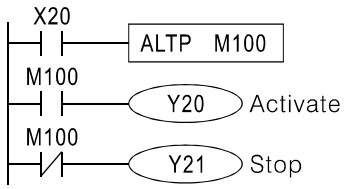


D : Destination Device

- When X20= "OFF " → "ON" for the first time, M0= "ON"; while X20= "OFF" → "ON" for the second time, M0= "OFF". Hence when X20= "OFF" → "ON" for an odd time, M0= "ON", and for an even time, M0= "OFF".

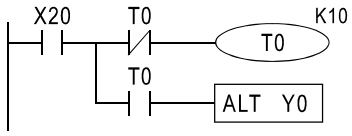


Activate and stop by operating the same push button control

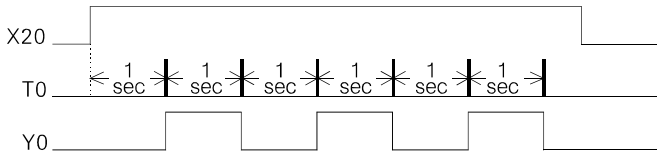


- When X20= "OFF" → "ON" for the first time, M100= "ON"; Y20= "ON", Y21= "OFF" and the operation will be activated.
- When X20= "OFF" → "ON" for the second time, M100= "OFF"; Y20= "OFF", Y21= "ON" and the operation will be stopped.

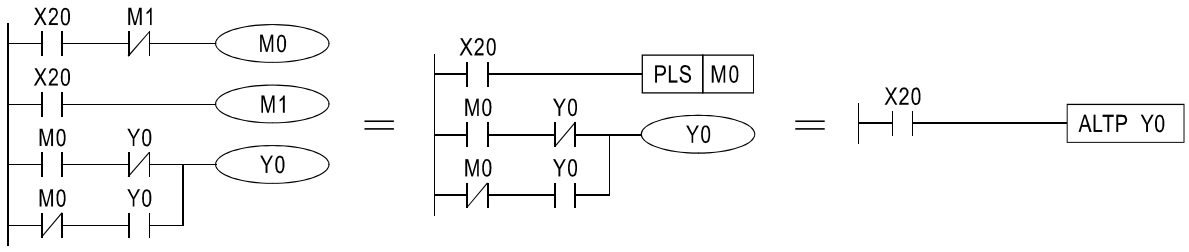
Produce the flashing state



- When X20= "ON", T0 will produce a pulse every other second.
- Every time when T0 produces a pulse, the state of Y0 will be changed once.



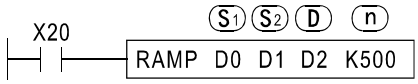
Traditional circuit : single-"ON"/double- "OFF"



FNC 67 RAMP	RAMP (S1) (S2) (D) (n)	Ramp Variable Value	M	VB	VH
			○	○	○

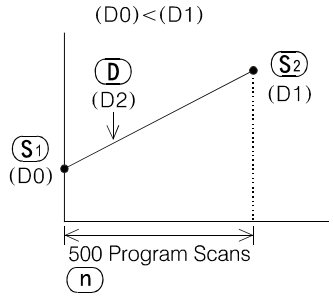
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○					○
S2											○					○
D											○					○
n															○	

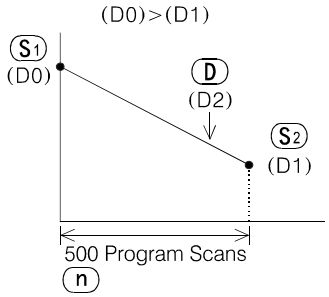
- n= 1 ~ 32767
- D occupies 2 consecutive registers



S1 : Initial value of the ramp signal
S2 : Destination value of the ramp signal
D : Value of the "journey" of the ramp signal
n : Specified number of Scan Times of the "journey" takes

- Write the initial point value of the ramp signal to D0 and write the destination value of the ramp signal to D1.
  When X20= "ON" and (D0) < (D1), then the current value of D2 will increase from the setting value of D0 to that of D1.
  When X20= "ON" and (D0) > (D1), then the current value of D2 will decrease from the setting value of D0 to that of D1.
  It takes 500 of PLC's Scan Times for the current value shifted from the setting value of D0 to that of D1.





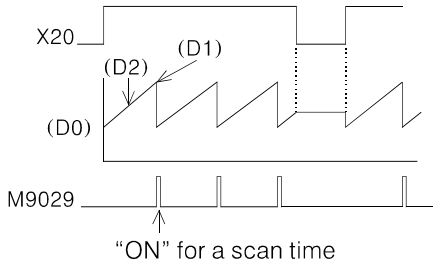
When the instruction is performed, the value of the "journey" of the ramp signal will be reflected on (D2) while the value of the "journey" of the scan times will be reflected on (D3).

- As shown in the diagram above, whether the pointing curve of D2, appears to be in Linear Gradient is closely correlated to the scan time of PLC. Generally PLC does not always take the same scan time. Thus, if in the occasion where the RAMP instruction is applied and it requires Linear Gradient, the interval that the RAMP instruction is performed must be equal each time. In terms of this purpose, it's acceptable to use the constant scan time setting function or the interrupt function. (Please reference to the program examples in next page.)
- When X20= "ON" → "OFF", the instruction will be disabled and D3 will be cleared as "0"; And if X20 is set "ON" again, the instruction will restore.
- When the execution of the instruction is completed, M9029= "ON" and the content value of D2 will be restored to the setting value of D0.
- The instruction can work with the analog output to incorporate the action of the buffered activation/stop.
- If X20= "ON" and PLC turns from STOP to RUN, please clear D3 as "0" (placed at the front end of the program).

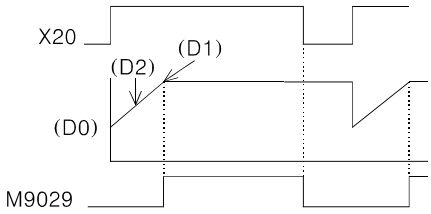
Operation Modes (swapped by Flag M9026)

When the RAMP instruction is performed, the operation mode will change depending on the status of Special Auxiliary Coil M9026.

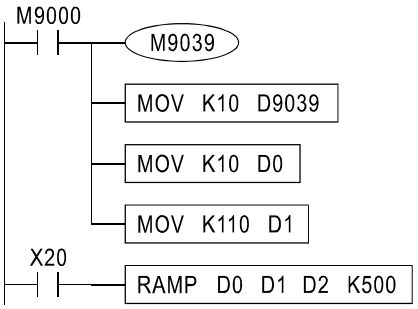
- It will generate contiguous ramp signals. (Repeat Mode)



- It will generate only one ramp signal. (Hold Mode)



A program model for usage of the constant scan time function



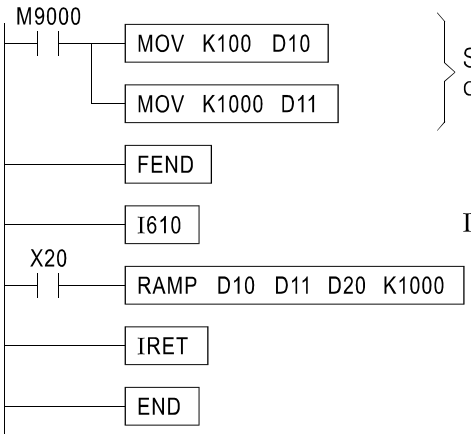
Settle the scan time at 10ms.

Set the initial value of the Ramp signal at 10 and the destination at 110.

Time of the entire journey of the RAMP instruction (the current value of D2 changes from the setting value of D0 to that of D1) will be 10ms × 500=5,000ms=5 Secs.

Since the scan time is settled at 10ms (each scan time is consistent) the ramp signal appears to be in Linear Gradient. However, in the program model referred above, it should be noted that the setting value of the constant scan time is required to be a little larger than the maximum value of the actual scan time. Otherwise, the constant scan time function would be useless. To oversee the D9012 register will get the maximum value of the actual scan time.

A program model for usage of the Time Interrupt function



Set the initial value of the Ramp signal at 100 and the destination at 1000.

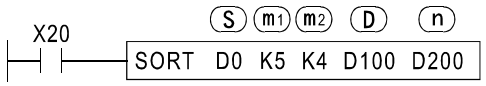
I610 is the interrupt pointer of the time interrupt at 10ms.

Time of the entire journey of the RAMP instruction (the current value of D20 changes from the setting value of D10 to that of D11) will be 10ms × 1000=10,000ms=10 Secs.

FNC 69 SORT	SORT (S) (m1) (m2) (D) (n)	Sort Data	M	VB	VH
			○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					
m1															○	
m2															○	
D											○					
n											○				○	

● m1=1 ~ 32 ● m2=1 ~ 6 ● n=1 ~ m2



S : Head register ID number of the original data block

m1: Number of data records to be sorted

m2: Number of data fields of each set

D : Head register ID number of the data block where Sort results are stored

n : Reference value of Sort data

Original Data Table
(Start from destination register (S))

	Data Filed			
	1	2	3	4
	Student ID	Language	Mathematics	History
(m1)	(D0) 1	(D5) 80	(D10) 70	(D15) 75
	(D1) 2	(D6) 65	(D11) 70	(D16) 90
	(D2) 3	(D7) 90	(D12) 65	(D17) 80
	(D3) 4	(D8) 75	(D13) 90	(D18) 65
	(D4) 5	(D9) 80	(D14) 85	(D19) 95
	(m2)			

- The SORT instruction is used to sort several data records (designated by (m1)). Each may have some data fields (the number of data fields is designated by (m2)) while "(n)" is used to assign the nth field as the basis for Sort Data. (S) designates the head register ID number of the original data to be sorted and (D) designates the head register ID number of the data block where Sort results are stored.
- When X20= "ON", the Sort instruction is performed. This instruction completes the Sort action only after m1 scan cycle(s). When the Sort is completed, the Execution Completed Flag M9029="ON" for a can time and the Sort action will be stopped.
- Both (S) and (D) will occupy (m1)×(m2) consecutive register(s)
- The SORT instruction can be used once only in the program.

Sort Data Result Table
(Start from destination register (D))
when D200=2

	Data Filed			
	1	2	3	4
	Student ID	Language	Mathematics	History
(m1)	(D100) 2	(D105) 65	(D110) 70	(D115) 90
	(D101) 4	(D106) 75	(D111) 90	(D116) 65
	(D102) 1	(D107) 80	(D112) 70	(D117) 75
	(D103) 5	(D108) 80	(D113) 85	(D118) 95
	(D104) 3	(D109) 90	(D114) 65	(D119) 80
	(m2)			

Sort Data Result Table
(Start from destination register (D))
when D200=4

	Data Filed			
	1	2	3	4
	Student ID	Language	Mathematics	History
(m1)	(D100) 4	(D105) 75	(D110) 90	(D115) 65
	(D101) 1	(D106) 80	(D111) 70	(D116) 75
	(D102) 3	(D107) 90	(D112) 65	(D117) 80
	(D103) 2	(D108) 65	(D113) 70	(D118) 90
	(D104) 5	(D109) 80	(D114) 85	(D119) 95
	(m2)			

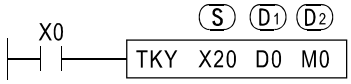
MEMO

6-9 External Setting and Display Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
70	D	TKY		Ten Key input	○	○	
71	D	HKY		Hexadecimal Key input	○	○	
72		DSW		Digital Switch (thumbwheel input)	○	○	
73		SEGD	P	Seven Segment Decoder	○	○	○
74		SEGL		Seven Segment with Latch	○	○	
76		ASC		ASCII code Convert	○	○	
77		PR		Print	○	○	
78	D	FROM	P	Read from a special function block	○	○	
79	D	TO	P	Write to a special function block	○	○	

D	FNC 70 TKY		DTKY (S) (D1) (D2)	Ten Key Input	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S	○	○	○	○												○
D1						○	○	○	○	○	○					○
D2		○	○	○												○

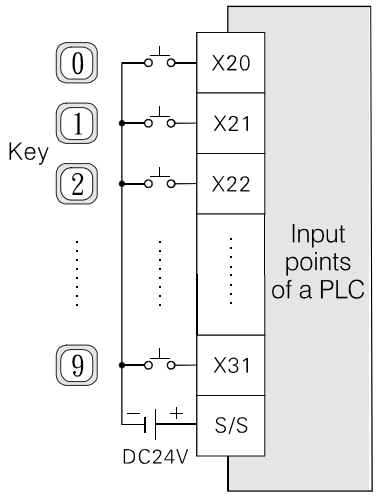


S : Initial device of the key input

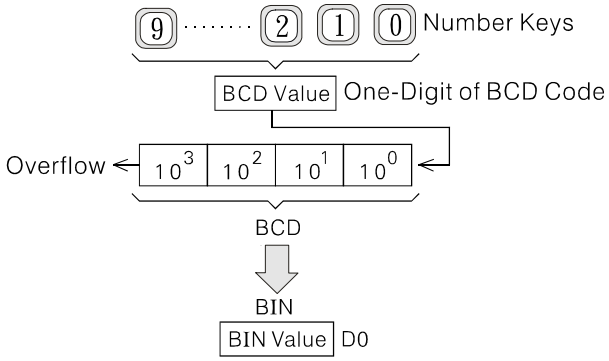
D1 : Place where the key input value is stored

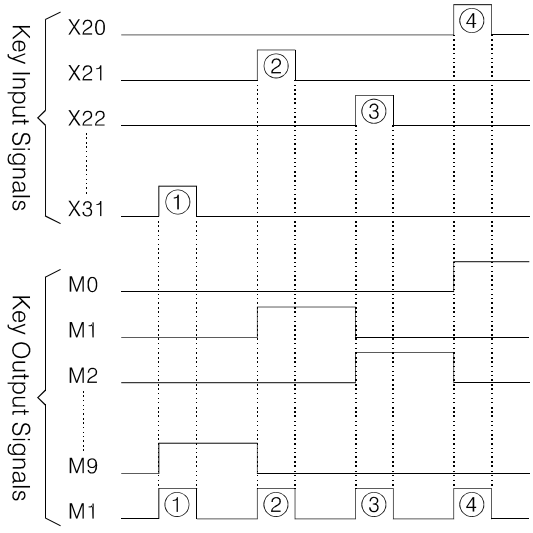
D2 : Initial destination device of the key output signal

- The instruction designates consecutive ten input devices, initiating from (S), which represent decimal numbers 0 ~ 9 in order. These 10 external input devices are connected to 10 keys. Based on the 10 keys pressed is sequence, a four-digit decimal number 0 ~ 9,999 (a 16-bit instruction) or an eight-digit decimal number 0 ~ 99,999,999 (a 32-bit instruction) can be input. And then, the input value will be placed in (D1). Also the instruction uses 11 consecutive devices which starting from (D2) to store the status of the keys.



- As in the left diagram, the input points X20 ~ X31 are received the signals from keys 0 ~ 9. When X0 = "ON", the instruction started to be performed. The value input by the number keys is placed in D0 (by format of BIN) and the statuses of the keys are restored in M0 ~ M9 and M10.

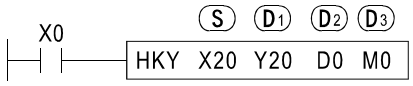




- As shown in the left sequence diagram, the number keys and following X20 ~ X31 are input in order, then the result 9,120 is stored in D0.
- When X31 is connected (key #9 is pressed), M9 will turn "ON" and remain "ON", until the next key is pressed (X21 = "ON" → M9 = "OFF"). The same situation applies to other keys.
- If any of the keys X20 ~ X31 is pressed, M10 = "ON" and the devices corresponding to M0 ~ M9 are "ON".
- When the status of X0 "ON" → "OFF", the value of D0 will stay unchanged and M0 ~ M10 will all turn "OFF".

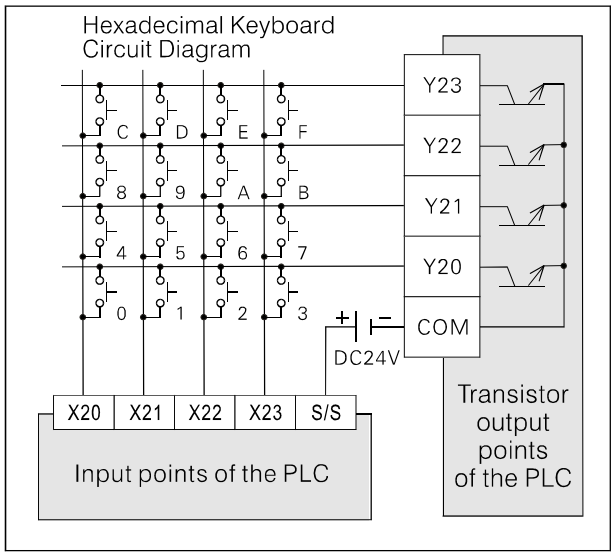
D	FNC 71 HKY		DHKY (S) (D1) (D2) (D3)	Hexadecimal Key Input	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S	○															○
D1		○														○
D2									○	○	○					○
D3		○	○	○												○



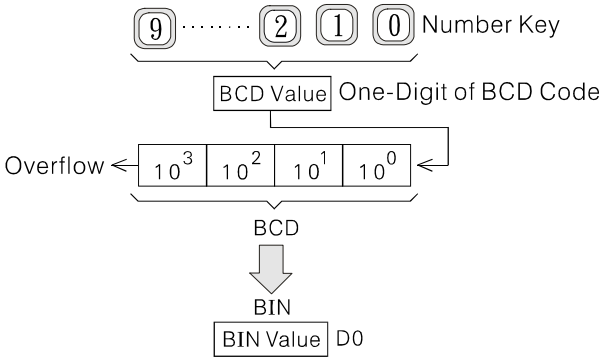
S : Multiplex scanning initial point of the key input
D1 : Multiplex scanning initial point of the key output
D2 : Place where the key input value is stored
D3 : Initial destination device of the key output signal

- The instruction creates the Hexadecimal Key Input by matrix scan of 4 consecutive external input points initiating from (S) and 4 consecutive external output points initiating from (D1). The value input by the number key is stored in (D2). And the instruction uses 8 consecutive devices which starting from (D3) to store the status of the keys.



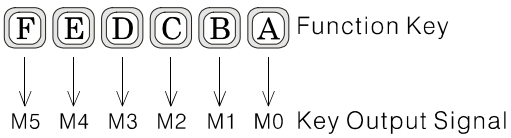
- In the left diagram, the Hexadecimal Keyboard is composed of X20 ~ X23 and Y20 ~ Y23. When X20 = "ON", the instruction started to be performed. The value input by any key is placed in D0 and the status of the key is restored in M0 ~ M7.
- M9029 will turn "ON" for a scan time when the instruction is performed for a scan.
- If there are several keys being pressed at the same time, only the key activated first is effective.
- If the special coil M9167 is already "ON", the HKY instruction can be used for input a hexadecimal value 0 ~ F.
- The HKY instruction can be used once only in the program.
- This instruction is only recommended for use with transistor output modules.

Number Input



- The value input by the number keys is placed in D0 (by format of BIN).
- For a 16-bit instruction, the available maximum input value will be 9,999. If an input value has more than 4 digits, the previous digit(s) will be overflowed.
- FOr a 32-bit instruction, the available maximum input value will be 99,999,999. If an input value has more than 8 digits, the previous digit(s) will be overflowed.

Number Input



- The A ~ F keys are defined as function keys.
- If a function key is pressed, the corresponding key output signal will turn "ON" and remain the same status, until other function key has been pressed the previous signal will be "ON" → "OFF".

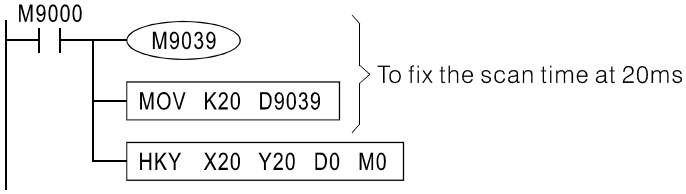
  For example, when [A] is pressed, M0 will turn and remain "ON". And if [F] is pressed then, M5 will turn and remain "ON" while M0= "OFF".

Key Output Signal

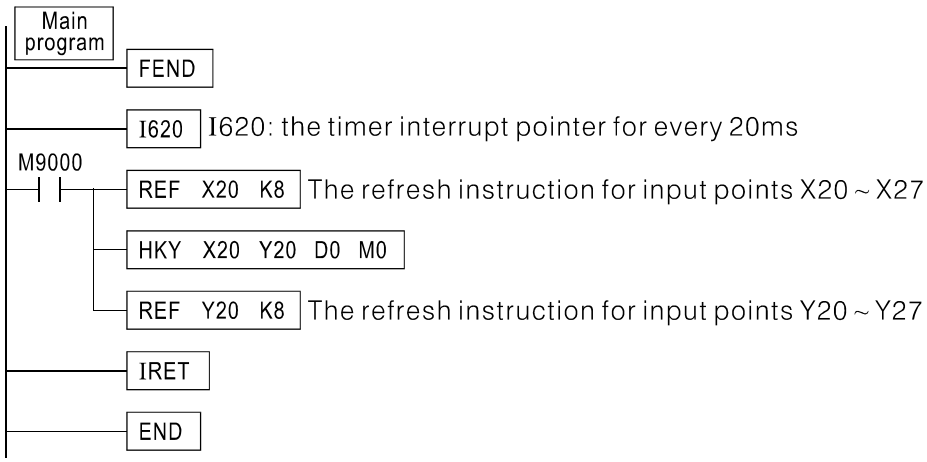
- If the keys [A] ~ [F] are pressed, the corresponding key output signals M0 ~ M5 will turn "ON".
- During the period when any one of the function keys [A] ~ [F] is pressed, M6= "ON"; And M6= "OFF" when the key is released.
- During the period when any one of the number keys [0] ~ [9] is pressed, M7= "ON"; And M7="OFF" when the key is released.
- When the conditional contact X0= "OFF", the input value will stay unchanged; However, M0 ~ M7 will all turn "OFF".

Notice

- When the instruction is performed, it should take 8 scan times to effectively capture a key. When the program's scan time is too long or too short, it may affect to read the key input signal incorrectly. The solution may be shown as follows:
- If the scan time is too short, this may possibly does not have enough time to take the I/O responses then it will cause to read the input keys incorrect. Please use the constant scan time function to fix the scan time at 20ms.

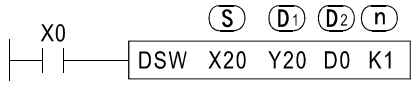


- If the scan time is too long, this will cause key responses to be slow. Please use the timer interrupt function to fix the scan time of keys at 20ms.



FNC 72 DSW	DSW (S) (D1) (D2) (n)	Digital Switch Input (Thumbwheel Input) Input	M	VB	VH
			○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S	○															○
D1		○														○
D2									○	○	○					○
n															○	
• n=1～2																



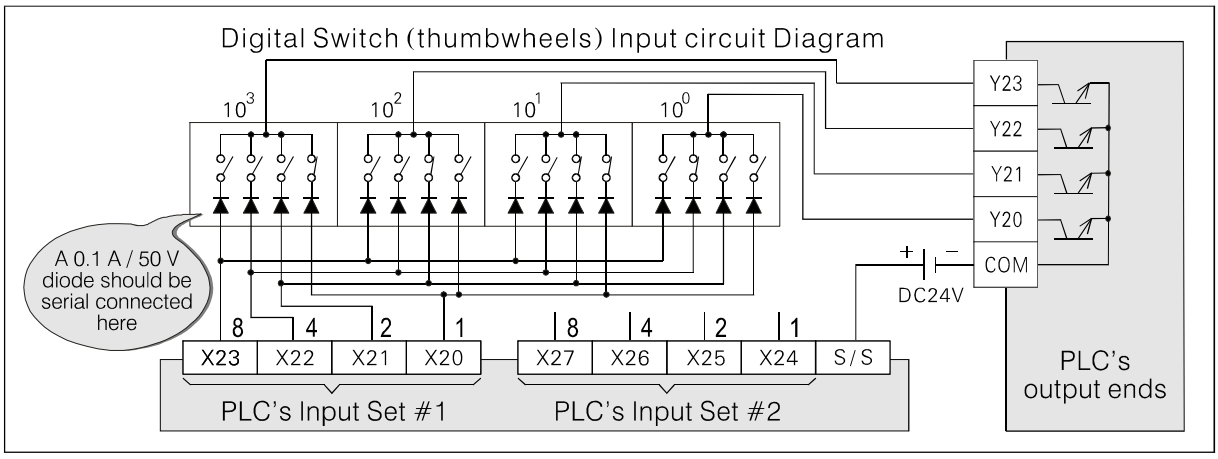
S : Multi-scan the digital switch input of initial point

D1 : Multi-scan the digital switch output of initial point

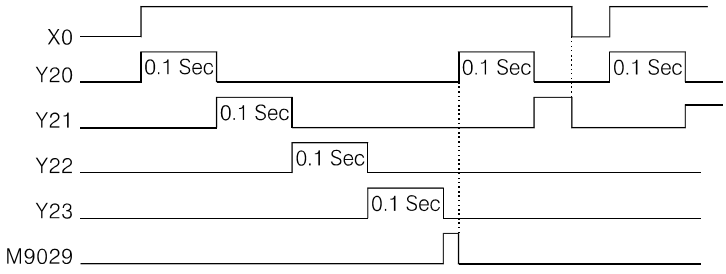
D2 : Place the value of the digital switchs

n : Number of digital switch sets connecting

- The instruction scans and reads one set (or two sets) of four-digit thumbwheel digital switches by 4 (or 8) consecutive input points initiating from (S) and 4 consecutive output points initiating from (D1). The value of the digital switch is stored in (D2) to read 1 or 2 sets of four-digit switches is decided by (n).

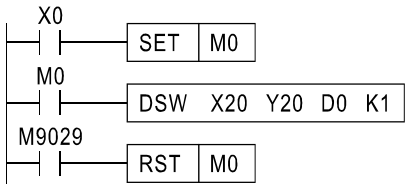


- The diagram shown above is a circuit of a multiplexed thumbwheels switch which is composed by X20～X23 and Y20～Y23. When X0= "ON", the instruction will start to be performed, the value of the thumbwheels will be read and converted into a BIN format then the value will be stored in D0. If (n)=k2 and input points X24~X27 are connected with another set of the thumbwheels then the value of the 2nd set of the thumbwheels will be stored in D1.



- The left diagram is a scan sequence diagram. When X0= "ON", Y20～Y23 will automatically cycle the scan. If each cycle is completed, Execution Completed Flag M9029 will be "ON" for a scan time.
- The instruction can be used only once in the program and recommended to use transistor output unit(s) for the multiplex scan output ends Y20～Y23.

The approach using the relay output unit(s) as the scan output end.



- X0 uses the push-button switch.
- The DSW instruction will read the value of the thumbwheel digital switch once when X0 is pressed once. In the remaining time, no DSW instruction would be performed, nor does any scan action proceed. Therefore, there's no problem even if the relay is used at the scan output end.

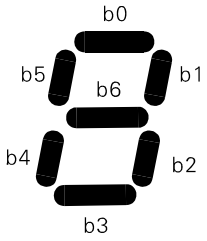
FNC 73 SEGD	P	SEGD P (S) (D)	Seven Segment decoder	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

S : Source device to be decoded

D : Output device after decoded

- When X20="ON", decode the content value (nibble format 0 ~ F) of D0's lower four bits (b3 ~ b0) into a code for a seven-segment display and output it through Y20 ~ Y27.
- The output structure of SEGD is shown in the following table.

(S)		Composition of the seven segment display	(D)								Data Displayed
Hexadecimal Number	Bit Format		b7	b6	b5	b4	b3	b2	b1	b0	
0	0000		0	0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		0	0	1	0	0	1	1	1	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		0	1	1	0	1	1	1	1	9
A	1010		0	1	1	1	0	1	1	1	A
B	1011		0	1	1	1	1	1	0	0	b
C	1100		0	0	1	1	1	0	0	1	C
D	1101		0	1	0	1	1	1	1	0	d
E	1110		0	1	1	1	1	0	0	1	E
F	1111		0	1	1	1	0	0	0	1	F

FNC 74 SEGL	SEGL (S) (D) (n)	Seven Segment with Latch	M	VB	VH
			○	○	

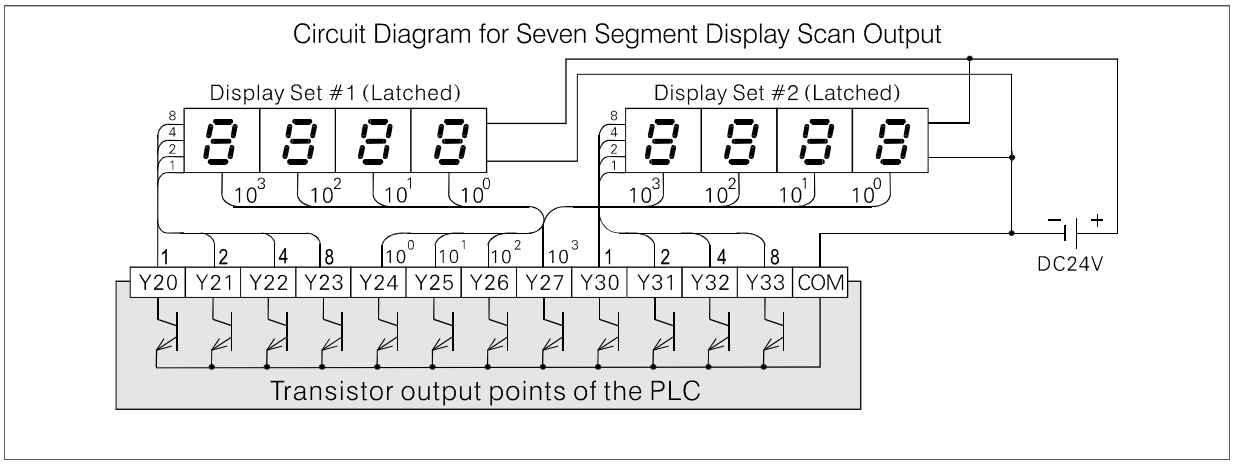
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D		○														○
n															○	
• n=0 ~ 7																

S : Source decimal value to be shown in the seven segment display

D : Initial point for the scan output of the seven segment display

n : Polarity designated for output signals and latch signals

- This instruction scan outputs to one (two) set(s) of four-digit seven segment display will occupy eight (Twelve) consecutive output points initiating from (D) and demonstrates the content value of (S) on the seven segment display. Whether there is one or two sets of four-digit display for the scan output is determined by "(n)", and also "(n)" is used to designate the polarity combination for the PLC output terminal and the display input terminal.



- The diagram shown above is the circuits of a seven segment display composed of Y20 ~ Y27. When X20= "ON", the instruction will start to be performed. The value of D0 will be converted into a BCD code then transferred and displayed in Set #1. If the value of D0 exceeds 9,999, an operation error will occur. If Display Set #2 is also connected with the circuit and the "(n)" value is set properly, the content value of D1 will be demonstrated on Display Set #2.
- when X20= "ON", Y24 ~ Y27 will cycle the output scan automatically. It takes 12 program scan times for a display cycle and M9029 will turn "ON" for a program scan time when each cycle is completed.

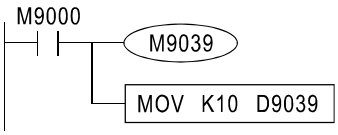
Setting value of "(n)"

A correct setting of "(n)" value is not only can be used to match the logic polarity of the PLC transistor output terminal with the input terminal of the seven segment display module but also to demonstrate there is one or two sets of display to be used.

Number of Display Sets	One set				Two sets			
Polarity of the PLC output terminal and the input terminal of the display data	Same		Different		Same		Different	
Polarity of the PLC output terminal and the input terminal of the display latched signal	Same	Different	Same	Different	Same	Different	Same	Different
n	0	1	2	3	4	5	6	7

The value of "n" is selected by referring to the table above, also it can use a number 0 ~ 3 or 4 ~ 7 to insert "n" orderly. And then test them one by one, until the value of the seven segment display is correctly demonstrate.

Notice



When the instruction is performed, at least it needs a 10ms of scan time. If the scan time is less than 10ms, please use the constant scan time function to fix the scan time at 10ms.

- The SEGL instruction can be used once only in the program.
- This instruction is only recommended for use with transistor output modules.

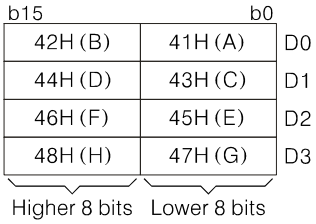
FNC 76 ASC	⊣ ⊢ ASC (S) (D)	ASCII Code Convert	M	VB	VH
			○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S	Key-in eight English letters from computer															
D									○	○	○					○

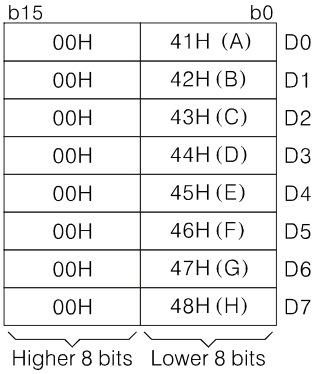
S : The source of English letters will be converted to ASCII codes

D : The device where ASCII codes are stored

- When X20= “ON”, English letters A ~ H will be converted into ASCII codes and stored in D0 ~ D3.



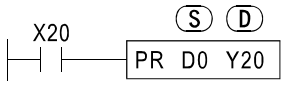
- If M9161= “ON”, each English letter will take over a register position after conversion into an ASCII code, where lower 8 bits (b7 ~ b0) of the register will store ASCII codes and higher 8 bits (b15 ~ b8) will be filled with zero (“0”).



- If the English letters contents in (S) is less than 8 characters, the difference is made up with “Space Key” Char (ASCII code 20H).

FNC 77 PR	PR (S) (D)	Output ASCII codes	M	VB	VH
			○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S									○	○	○					○
D		○														○
• D occupies 10 consecutive points																



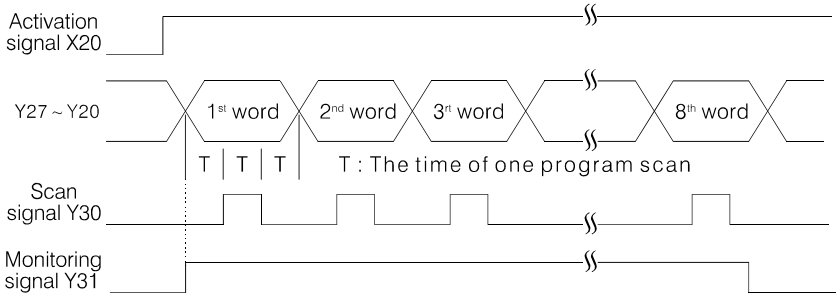
S : Source devices where ASCII codes are stored

D : Output points exporting ASCII codes

- The instruction will read ASCII codes of 4 (or 8) source registers (initiated from (S)) byte by byte. And then, orderly output the ASCII codes to the designated consecutive 8 output points (initiated from (D)).
- The process referred above designates the points from Y27 (the first bit) to Y20 (the last bit) are the data output points. It also designates Y30 as the scan signal and Y31 as the monitoring signal.
- There are two operation modes for the PR instruction, depending on the status "ON"/"OFF" of M9027.

M9027= "OFF"

- To generate the 8 words of sequence outputs. The operation sequence diagram is shown below:

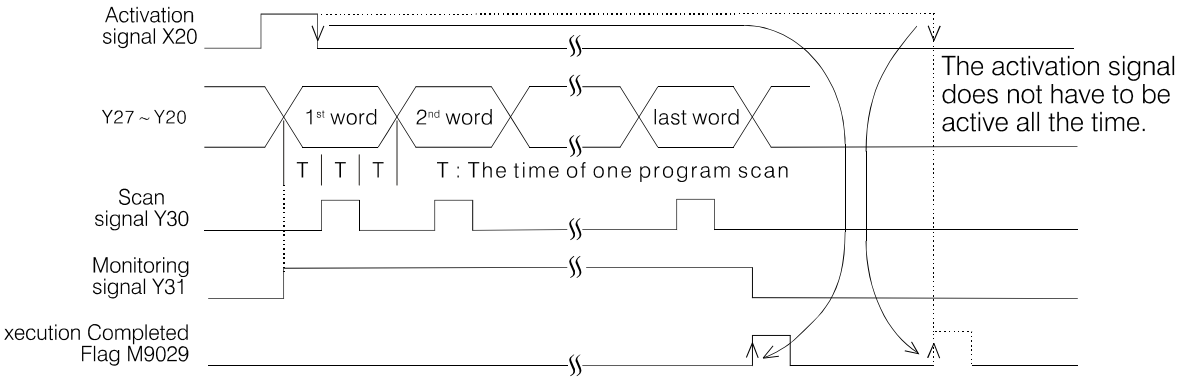


(S)	Up 8 bits	Down 8 bits
D0	2nd word	1st word
D1	4th word	3rd word
D2	6th word	5th word
D3	8th word	7th word

- If X20 turns "OFF" during the instruction is performed, the instruction is disabled then the data output will be discontinued. When X20 turns "ON" again, data will be transferred from the first letter.

M9027= "ON"

- To generate the 16 words of sequence outputs. The operation sequence diagram is shown below:

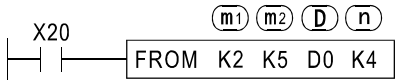


- The code "00H" (NUL) represents the end of the string and the following words will not be processed.
- If X20 always stays "ON", the output will be stopped automatically when all data are finished. Meanwhile M9029 will not be activated until X20 turns "OFF".

- Please use a transistor output unit for the output points designated by the instruction.
- The PR instruction can be used once only in the program.
- When performing the instruction, please use the constant scan time function to fix the scan time or place the instruction in a subroutine of the timer interrupt function, they will fix the time value of "T" which shown in the diagram above.

D	FNC 78 FROM	P	DFROMP (m1) (m2) (D) (n)	Read special module BFM	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
m1															○	
m2															○	
D						○	○	○	○	○	○					○
n															○	
• M Series: m1=1 ~ 31 • VB0 Series m1=1 ~ 4 ; VB1 Series m1=1 ~ 8 ; VB2 Series m1=1 ~ 16																
• m2=0 ~ 32767 • n=1 ~ 32767																



m1 : The position number of specified special module

m2 : Initial serial number of the BFM to be read

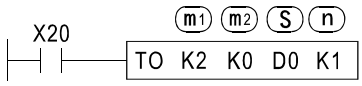
D : The initial device of storage for collect the picked up data

n : Number of data groups to be read

- The CPU module of M Series and the Main Unit of VB Series PLC use the instruction to read BFM data of the special module.
- When X20= "ON", 4 groups (they will be BFM #5 ~ BFM #8 because (n) =4 and (m2) =5) data in the specified special module (which is installed in the (m1) =2nd position) will be read and stored in D0 ~ D3.
- About the special module of the M Series and VB Series, the definitions of position are different. To assign the (m1) of M Series, please refer to the next page; For the (m1) of VB Series, each special module is consecutively assigned from K1 to K16, it begins with the closest one to the Main Unit.
- When X20= "OFF", the instruction will not be performed but the data (which was read previously) will still remain.

D	FNC 79 TO	P	DTOP (m1) (m2) (S) (n)	Special module BFM write in	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
m1															○	
m2															○	
S					○	○	○	○	○	○	○				○	○
n															○	
• M Series: m1=1 ~ 31 • VB0 Series m1=1 ~ 4 ; VB1 Series m1=1 ~ 8 ; VB2 Series m1=1 ~ 16																
• m2=0 ~ 32767 • n=1 ~ 32767																



m1 : The position number of specified special module

m2 : Initial serial number of BFM which will be written

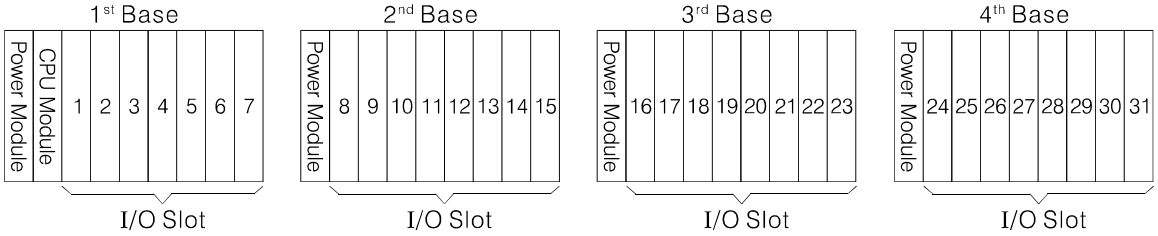
S : The initial source device, which stores the data is for the BFM

n : Number of data groups to be write

- The CPU module of M Series and the Main Unit of VB series PLC use the instruction to write BFM data to the special module.
- When X20= "ON", the content value of D0 will be written into BMF #0 of the special module which is installed in the 7th position. Because n=1, there is only one data group written in.
- About the special module of the M Series and VB Series, the definitions of position are different. To assign the (m1) of M Series, please refer to the next page; For the (m1) of VB Series, each special module is consecutively assigned from k1 to k8, it begins with the closest one to the Main Unit.
- When X20= "OFF", the instruction will not be performed but the data (which was written into the BFM previously) will still remain.

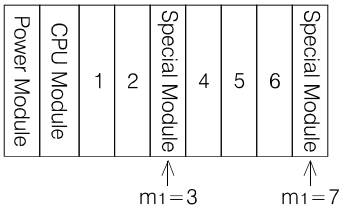
Number (m1) of the Slot where Special Module is Located (For M Series only)

- M Series PLC is a module structure programmable controller. The system is composed of various I/O modules and installed on the base unit. M Series can be connected up to 4 bases and the I/O slot number is shown in the following diagram:

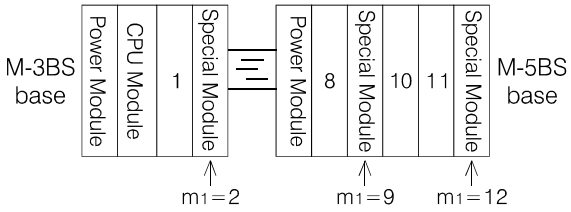


- The (m1) operand "number of the slot where the special module is located" of the FROM/TO instruction is a location referring to the diagram above. In the following M-3BS is a base with 3 I/O slots, M-5BS is a base with 5 I/O slots and M-8BS is a base with 8 I/O slots.

Example 1: Only an M-8BS base



Example 2: An M-3BS base connected with an M-5BS base

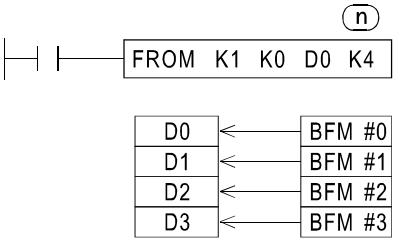


BFM Number (m2)

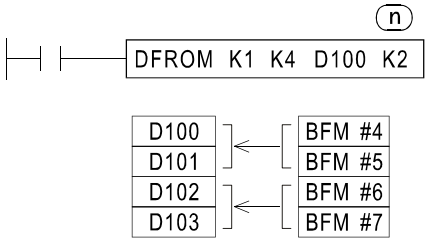
- M Series and VB Series PLC contain the Buffer Memory (BFM) which is used to store the setting value of the special module and various operation statuses. Each BFM data register has a length of 16 bits, and different special modules have different numbers of BFM registers. The number of BFM register is coded in decimal, such as #0, #1,....#9, #10,.....
- If a module is used the BFM to transfer data between itself and the Main Unit, it is called the Special Module.

Number of Data Groups (n) Transferred

- 16-bit instruction



- 32-bit instruction



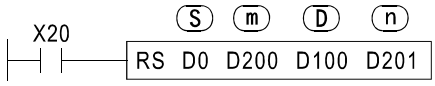
- The number of the data groups transferred is determined by "(n)". n=4 in a 16-bit instruction has the same meaning with n=2 in a 32-bit instruction.

6-10 Serial Communication Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
80		RS		Serial Communications Instruction	○	○	○
81	D	PRUN	P	Parallel Run	○	○	
82		ASCI	P	Converts HEX to ASCII	○	○	○
83		HEX	P	Converts ASCII to HEX	○	○	○
84		CCD	P	Check Code	○	○	○
85		VRRD	P	VR Volume Read	○	○	○
86		VRSC	P	VR Volume Scale	○	○	○
89		LINK		Easy Link Communication	○	○	

FNC 80 RS	RS (S) (m) (D) (n)	Serial communications Instruction	M	VB	VH
			○	○	○

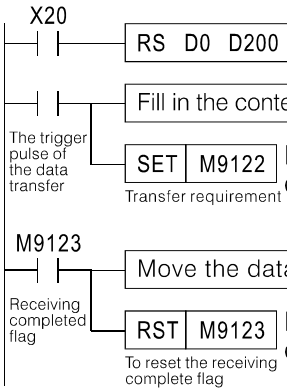
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
m											○				○	
D											○					○
n											○				○	
• m,n=0～256																



S : Head ID number of the register transferring data
m : Number of groups transferring data
D : Head ID number of the register receiving data
n : Number of groups receiving data

- When M Series PLC's M1-CPU1 module is equipped with the communication expansion card M-232R or M-485R, therefore this CPU module is provided with the CP2 (the second Communication Port). Then the instruction can be used to transfer or receive the data via the serial communications interface of external peripheral facilities.
- When VB or VH Series PLC's Main Unit is equipped with the communication expansion card (VB-232 or VB-485) or expansion module (VB-485A , VB-CADP etc.), therefore this CPU module is provided with the CP2 (the second Communication Port). Then the instruction can be used to transfer or receive the data via the serial communications interface of external peripheral facilities.
- The CP2 is a multi-functional expansion communication port, it can operation various communication types. When the CP2 is assigned to this instruction, the manage type should select to "Non protocol". About the CP2, to select the manage type and related parameter setting, please specify it from the programming software (Ladder Master - System - 2nd COM Port Setting).
- Designate "(m)" as K0 where data transmission (send) is not needed, and designate "(n)" as K0 where data received is not needed.
- As many commercialized peripheral facilities (e.g. Inverters, barcode readers, card readers, electronic displays, etc.) equipped with serial communications interface have their individual protocols, PLC users have to use the RS instruction writing communication programs (in accordance with the communication protocol format of peripheral facilities), when M series PLC is to be connected with peripheral facilities, to transfer data between PLC and those peripherals.
- If the communication of the RS instruction is performed, data transmissions can be divided into 16-bit mode (M9161= "OFF") and 8-bit mode (M9161= "ON").
- M9063 will turn "ON" when any error occurs during data transmissions and receiving and the error code will store in D9063.
- More than one RS instruction can be programmed but only one may be active at any one time.

Sequence of Data Transmissions and Receiving



Write the data to be transferred, which headed with D0 and the number of groups is specified by D200.

M9122 will turn "OFF" automatically when data transmissions are completed. Do not use the RST instruction to reset the M9122.

Move the received data, which headed with D100 to the data storage area.

Reset the Receive Completed flag M9123 and prepare to receive the coming data. Do not reset M9123 consecutively in the program.

Related Flags and Data Register

① Transmission Trigger Flag M9122

- When the conditional contact X20= "ON", the RS instruction is performed. At this time, if the pulse signal forces the status of M9122 to be "ON", the content value of the register initiating from D0 will be transferred via the serial interface. When the data transmission is completed, M9122 will be reset to "OFF" automatically.

② Receive Completed Flag M9123

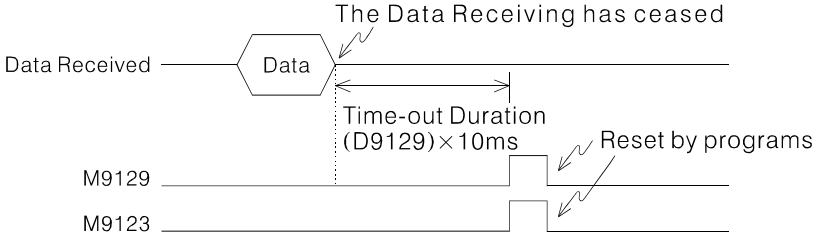
- When the conditional contact X20= "ON", the RS instruction is performed. PLC is ready for the status of receiving.
- When the data receiving is completed, M91213= "ON". At this moment, the received data in the buffer will be moved to the data storage area, and then M9123 will be reset to "OFF". Afterwards, PLC will be ready for the status of receiving immediately.

③ Carrier Detection Flag M9124 (the VH series does not support this flag)

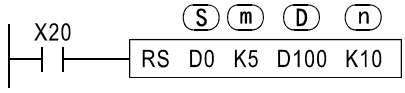
- When PLC receives the CD (Carry Detect) signal from the serial interface, M9124= "ON".
- When PLC is connected with a MODEM, the CD signal is used to represent the status of MODEM. If M9124= "OFF", the transmission of the dialing signal can be performed. If M9124= "ON", data transmissions and receiving can be performed.

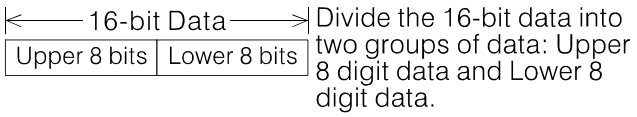
④ Time-out Flag M9129

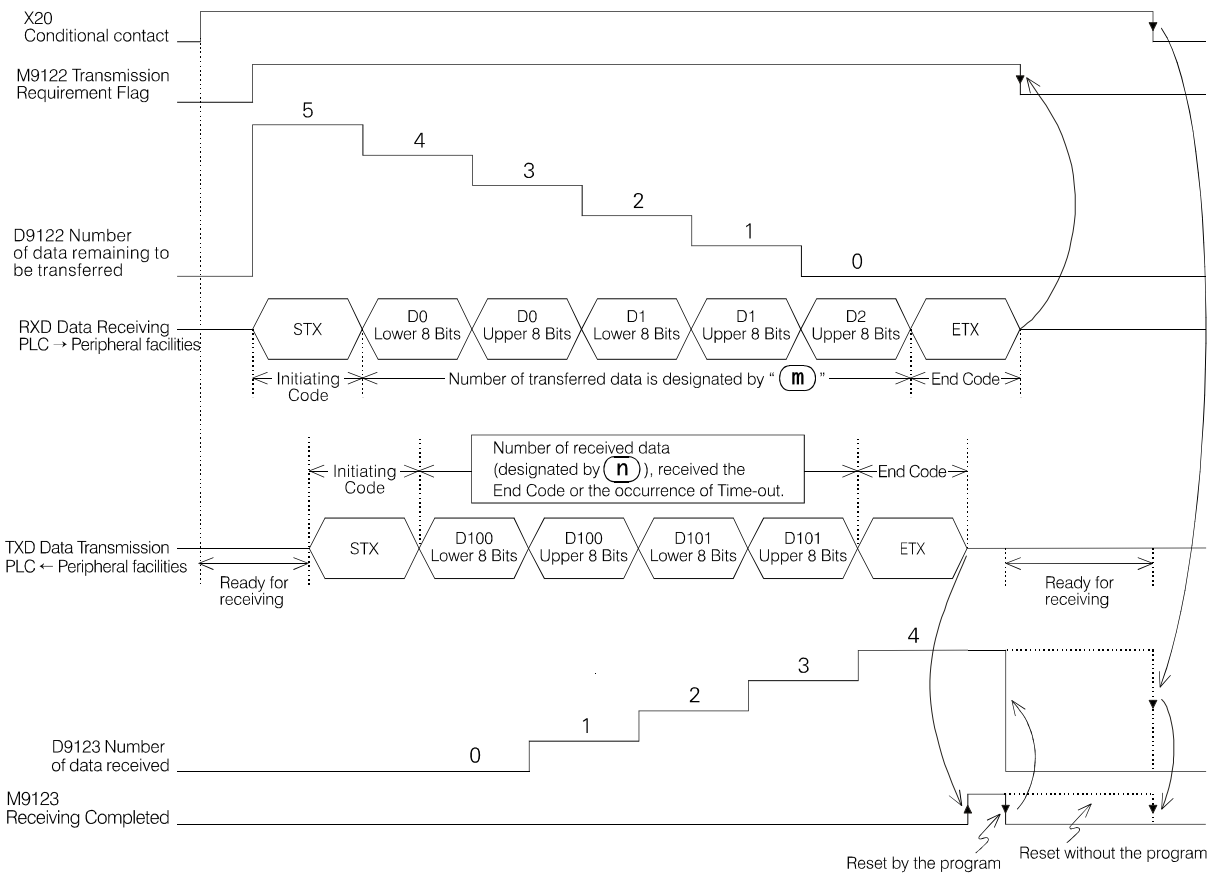
- During the data receiving, if the receiving time exceeds the time-out duration (designated by D9129), M9129 will turns "ON" to represent as the occurrence of Time-out, and also the Receive Completed flag M9123 will be forced "ON" to close the data receiving action.
- The M9129 will not be reset automatically, must using an instruction in the program to reset the status of M9129.
- By applying the Time-out function, PLC will receive the data of transferred from peripheral facilities which is no "End Code" or no length can be predicted.
- The setting value of the Time-out duration is restored in D9129. The Time-out duration = (the content value of D9129) × 10ms. When D9129=0 (the default value), the Time-out duration is 100ms.



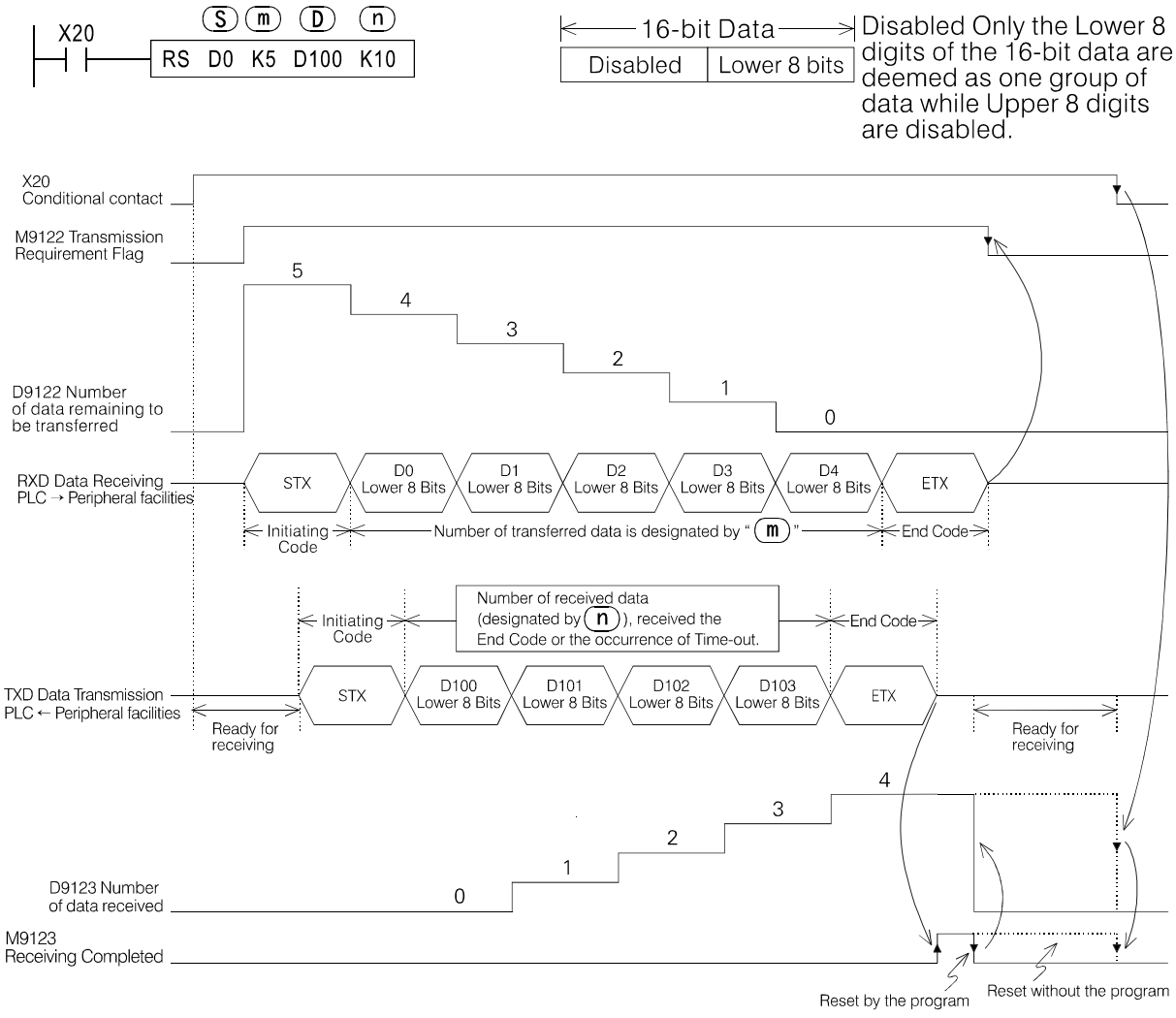
Description of Data Transmissions and Receiving Actions: 16-bit Mode (M9161="OFF")







Description of Data Transmissions and Receiving Actions: 8-bit Mode (M9161="ON")

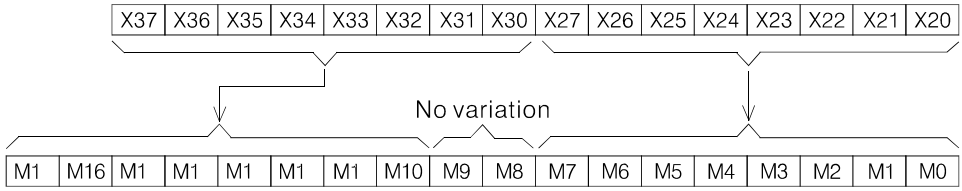


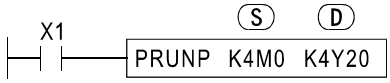
D	FNC 81 PRUN	P	DPRUNP (S) (D)	Parallel Run	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○		○									○
D						○	○									○
• The X, Y and M in the K*n*X, K*n*Y and K*n*M should assigned an ID number which the least digit is a zero "0".																
• When S=K*n*X, D should be K*n*M; And when S=K*n*M, D should be K*n*Y.																

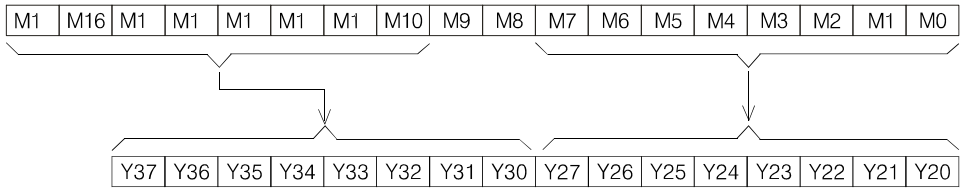
S : Transmission source devices
D : Transmission destination devices

- Transfer the source devices (Octonary number system, designated by (S)) to (D).
- When X0= "ON", transfer the content of K4X20 to K4M0 in Octonary number system.





- When X1="OFF"→ "ON", transfer the content of K4M0 to K4Y20 in Octonary number system.



FNC 82 ASCI	P	ASCIP (S) (D) (n)	Converts HEX to ASCII	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D									○	○	○					○
n											○				○	

- When S is designated to K*n*X, K*n*Y, K*n*M or K*n*S, it should be designated to K4X, K4Y, K4M or K4S.
- n= 1 ~ 256

S : Head ID number of data source

D : Head ID number of the position where conversion results are stored

n : The number of hexadecimal data characters is selected

- When the instruction is performed, it converts each HEX character of the source devices (designated by (S)) into ASCII codes and transfers they to the designated devices (D). The number of the converted characters is determined by (n).
- ASCII codes corresponding to HEX values 0 ~ F are shown in the following table:

HEX Value	0H	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	BH	CH	DH	EH	FH
ASCII Code	30H	31H	32H	33H	34H	35H	36H	37H	38H	39H	41H	42H	43H	44H	45H	46H

- When X20= "ON", the instruction converts the 8-digit HEX value in D0 and D1 to ASCII codes, and transfers to the designated registers which are headed by D100.
- The instruction has two operation modes depending on the status of M9161:
  Assume "(S)"
  (D0)=4567H
  (D1)=89ABH

M9161="OFF" (16-bit Conversion Mode)

- This mode will divide each designated device (D) into Upper 8 bits and Lower 8 bits, where two ASCII codes are stored respectively.

(D)	n=8	n=7	n=6	n=5	n=4	n=3	n=2	n=1 (n)
D100 Lower 8 Bits	38H	39H	41H	42H	34H	35H	36H	37H
D100 Upper 8 Bits	39H	41H	42H	34H	35H	36H	37H	
D101 Lower 8 Bits	41H	42H	34H	35H	36H	37H		
D101 Upper 8 Bits	42H	34H	35H	36H	37H			
D102 Lower 8 Bits	34H	35H	36H	37H				
D102 Upper 8 Bits	35H	36H	37H					
D103 Lower 8 Bits	36H	37H						
D103 Upper 8 Bits	37H							

M9161="ON" (8-bit Conversion Mode)

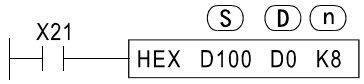
- This mode will divide each designated device (D) into Upper 8 bits and Lower 8 bits, while Upper 8 bits are filled with zero ("0") and Lower 8 bits store an ASCII codes, each register stores an ASCII code only.

(D)	n=8	n=7	n=6	n=5	n=4	n=3	n=2	n=1 (n)
D100	38H	39H	41H	42H	34H	35H	36H	37H
D101	39H	41H	42H	34H	35H	36H	37H	
D102	41H	42H	34H	35H	36H	37H		
D103	42H	34H	35H	36H	37H			
D104	34H	35H	36H	37H				
D105	35H	36H	37H					
D106	36H	37H						
D107	37H							

FNC 83 HEX	P	HEXP (S) (D) (n)	Converts ASCII to HEX	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D									○	○	○					○
n											○				○	

- When S is designated to K*n*X, K*n*Y, K*n*M or K*n*S, it should be designated to K4X, K4Y, K4M or K4S.
- n= 1 ~ 256



S : Head ID number of data source

D : Head ID number of the position where conversion results are stored

n : Number of ASCII codes converted

- When the instruction is performed, convert each ASCII code of the source device (which is designated by (S)) into a HEX value and transfer it to the designated devices (D). The number of ASCII codes converted is determined by (n).
- The following is a contrast table of ASCII codes and HEX values 0 ~ F:

ASCII Code	30H	31H	32H	33H	34H	35H	36H	37H	38H	39H	41H	42H	43H	44H	45H	46H
HEX Value	0H	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	BH	CH	DH	EH	FH

- When X21="ON", convert the ASCII code of the register initiating from D100 into a HEX value and transfer it to (D0) and (D1).
- If the content designated by Data source (S) is not an ASCII code of 0H ~ FH, PLC will regard it as an operation error and disable the instruction.
- The instruction has two operation modes depending on the status of M9161:

M9161="OFF" (16-bit Conversion Mode)

- This mode will convert the ASCII codes (stored in Upper 8 bits and Lower 8 bits) of each designated device (S) into HEX values.

(S)	
D100 Lower 8 Bits	38H
D100 Upper 8 Bits	39H
D101 Lower 8 Bits	41H
D101 Upper 8 Bits	42H
D102 Lower 8 Bits	34H
D102 Upper 8 Bits	35H
D103 Lower 8 Bits	36H
D103 Upper 8 Bits	37H

(D)								(n)
				0H	0H	0H	8H	n=1
				0H	0H	8H	9H	n=2
				0H	8H	9H	AH	n=3
				8H	9H	AH	BH	n=4
0H	0H	0H	8H	9H	AH	BH	4H	n=5
0H	0H	8H	9H	AH	BH	4H	5H	n=6
0H	8H	9H	AH	BH	4H	5H	6H	n=7
8H	9H	AH	BH	4H	5H	6H	7H	n=8
b15	Digit #3	Digit #2	b0	b15	位數3	位數2	b0	
		D1				D0		

M9161="ON" (8-bit Conversion Mode)

- This mode will convert the ASCII codes (stored in Lower 8 bits) of each designated device (S) into HEX values.

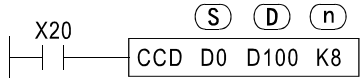
(S)	
D100 Lower 8 Bits	38H
D101 Lower 8 Bits	39H
D102 Lower 8 Bits	41H
D103 Lower 8 Bits	42H
D104 Lower 8 Bits	34H
D105 Lower 8 Bits	35H
D106 Lower 8 Bits	36H
D107 Lower 8 Bits	37H

(D)								(n)
				0H	0H	0H	8H	n=1
				0H	0H	8H	9H	n=2
				0H	8H	9H	AH	n=3
				8H	9H	AH	BH	n=4
0H	0H	0H	8H	9H	AH	BH	4H	n=5
0H	0H	8H	9H	AH	BH	4H	5H	n=6
0H	8H	9H	AH	BH	4H	5H	6H	n=7
8H	9H	AH	BH	4H	5H	6H	7H	n=8
b15	Digit #3	Digit #2	b0	b15	Digit #3	Digit #2	b0	
		D1				D0		

FNC 84 CCD	P	CCD P (S) (D) (n)	Check Code	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D									○	○	○					○
n											○				○	

- When S is designated to K*n*X, K*n*Y, K*n*M or K*n*S, it should be designated to K4X, K4Y, K4M or K4S.
- n= 1 ~ 256
- D occupies 2 consecutive points



S : Head ID number of Data source

D : Position where the result of SumCheck is stored

n : Number of data

- Sum up the content of n byte (8-bit) data headed with (S), total of the sum is stored in the designated device (D) while the Parity bits are stored in the next register.
- When the instruction is used for communication, the “SumCheck” (or “error detect code”) applied to ensure the accuracy of the data transmission.
- When X20= “ON”, sum up 8 consecutive 8-bit data headed with D0, total of the sum is stored in D100 while the Parity bits are stored D101.
- When X20= “ON”, sum up 8 consecutive 8-bit data headed with D0, total of the sum is stored in D100 while the Parity bits are stored D101.

M9161= “OFF” (16-bit Mode)

- This mode will take Upper 8 bits and Lower 8 bits of each device (designated by (S)) as an 8-bit data, and do the aggregate operation and generate the Parity data.

		Data Content value	MSB Content value in Binary							LSB
(S)	D0 Lower 8 Bits	255	1	1	1	1	1	1	1	1
	D0 Upper 8 Bits	80	0	1	0	1	0	0	0	0
	D1 Lower 8 Bits	135	1	0	0	0	0	1	1	1
	D1 Upper 8 Bits	28	0	0	0	1	1	1	0	0
	D2 Lower 8 Bits	100	0	1	1	0	0	1	0	0
	D2 Upper 8 Bits	73	0	1	0	0	1	0	0	1
	D3 Lower 8 Bits	210	1	1	0	1	0	0	1	0
	D3 Upper 8 Bits	5	0	0	0	0	0	1	0	1
(D)	D100	886								
	D101		1	1	0	0	1	1	1	0

When there is an odd number of “1”, the bit corresponding to D101 = 1.

When there is an even number of “1”, the bit corresponding to D101 = 0.

M9161=“ON” (8-bit Mode)

- This mode will take Lower 8 bits of each device (designated by (S)) as an 8-bit data (while ignore its Upper 8 bits), and do the aggregate operation and generate the Parity data.

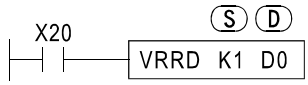
		Data Content value	MSB Content value in Binary							LSB
(S)	D0 Lower 8 Bits	255	1	1	1	1	1	1	1	1
	D1 Lower 8 Bits	80	0	1	0	1	0	0	0	0
	D2 Lower 8 Bits	135	1	0	0	0	0	1	1	1
	D3 Lower 8 Bits	28	0	0	0	1	1	1	0	0
	D4 Lower 8 Bits	100	0	1	1	0	0	1	0	0
	D5 Lower 8 Bits	73	0	1	0	0	1	0	0	1
	D6 Lower 8 Bits	210	1	1	0	1	0	0	1	0
	D7 Lower 8 Bits	5	0	0	0	0	0	1	0	1
(D)	D100	886								
	D101		1	1	0	0	1	1	1	0

When there is an odd number of “1”, the bit corresponding to D101 = 1.

When there is an even number of “1”, the bit corresponding to D101 = 0.

FNC 85 VRRD	P	VRRDP (S) (D)	VR Volume Read	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S															○	○
D						○	○	○	○	○	○	○		○		○
• S=1 ~ 2																

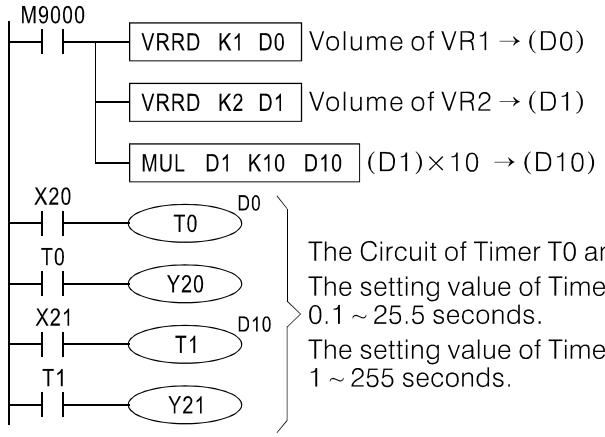


S : ID number of the Knob

D : Destination device where the volume is stored

- The VRRD instruction is used to read the volume of VR1 or VR2 in M series M1-CPU1 Module or VB series Main Unit. Convert the volume into a value ranging from 0 to 255 and store it in the designated device (D).
- When X20= "ON", convert the volume of VR1 into a BIN format which ranging from 0 to 255 and store it in D0.
- To acquire a value larger than 255, can multiply D0 by a constant.

M9161= "OFF" (16-bit Mode)



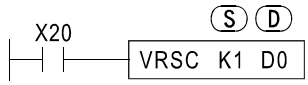
The Circuit of Timer T0 and T1

The setting value of Timer T0 is adjusted by VR1, the range will be 0.1 ~ 25.5 seconds.

The setting value of Timer T1 is adjusted by VR2, the range will be 1 ~ 255 seconds.

FNC 86 VRSC	P	VRSCP (S) (D)	VR Volume Scale	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S															○	○
D						○	○	○	○	○	○	○		○		○
• S=1 ~ 2																

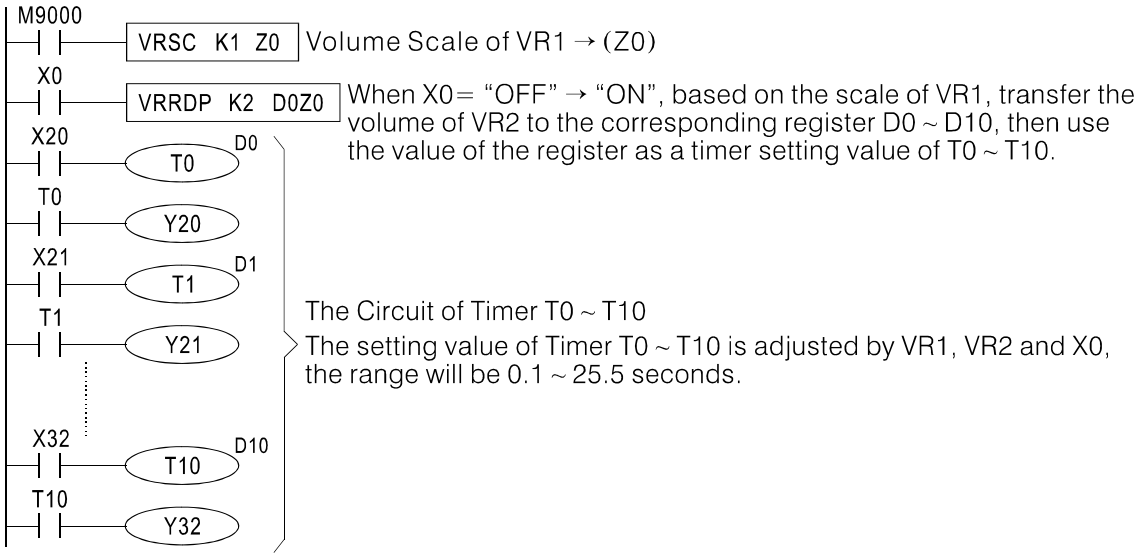


S : ID number of the Knob

D : Destination device where the volume is stored

- The VRSC instruction is used to read the scale of VR1 or VR2 in M series M1-CPU1 Module or VB series Main Unit. The scale (as a rotary switch with 11 set positions 0 ~ 10) is stored in the designated device (D). When the volume is located between two scales, it will rounds up or down to an integer 0 ~ 10.
- When X20= "ON", read the scale (0 ~ 10) of VR1 and store it in D0.

Using the value of VR1 and VR2 to change the timer setting value of T0~T10



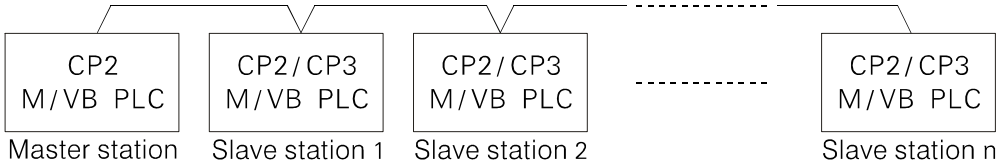
FNC 89 LINK	LINK (S1) (S2)	Easy Link Communication	M	VB	VH
			○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○					○
S2											○					

S1 : Head ID number of the register, which describe the data transfer/receive actions

S2 : Instruction working area, occupies 4 consecutive registers

- If the M Series CPU module mounts a M-232R or M-485 communication card, the CPU module will have the CP2 (2nd Communication Port). Then, via this instruction to proceed data transfer between PLCs.
- If the VB Series Main Unit mounts a communication card (VB-232 or VB-485) or a communication module (VB-485A, VB-CADP etc.), the Main Unit will have the CP2 (2nd. Communication Port). Then, via this instruction to proceed data transfer between PLCs.
- The CP2 is a multi-functional expanded communication port, it can be used for multiplex communication types. When the CP2 is assigned to this instruction, the communication type should use "EASY LINK" or "COMPUTER LINK". To select and relative parameters setting about the manipulation type of CP2, please use the option in the programming tool Ladder Master "System---2nd COM Port Setting..." to get the right setting.
- At most 256 nodes of M/VB Series PLC (slave VH series). can be linked together via this instruction and the RS-485 interface. The instruction can use for transfer the data of device X, Y, M, S, T, C and D.
- As the diagram below, select one of these linked PLCs as the Master station and the rest as Slave stations. Use the program develop devices (e.g. Ladder Master) to set the "EASY LINK" or "COMPUTER LINK" as the communication mode between the Master and Slave stations, and set each Slave station properly (the range of station ID number is 1 ~ 255). And then, write the data transmission/receiving command (designated by this instruction) to the Master station, to achieve the data transmission between PLCs.



- When X20= "ON", the LINK instruction will start to be performed. Based on the designated register string (which initiating from D1000), to do the data write or read action to the appointed Slave PLC station. And also, D100 ~ D103 store the status of the instruction execution.
- Every time the transmission/receiving operation which designated by (S1) is duly completed, the M9199 will be "ON" for a scan time. And then, it will repeat the data transmission/receiving processes from the first data again.
- When X20="ON" → "OFF", the instruction will be stopped and the data transmission/receiving will be disabled immediately.
- The LINK instruction can be used once only in the program.

- The register headed with (S1) is used to describe the data transmission/receiving information:

(S1)	Content Value	Description
D1000	1 ~ 255	To designate the number of transferred and received data sets. Each data transmission/receiving set should be described with 7 registers.
D1001	1 ~ 255	Designates the Slave station ID number, to proceed data transmission/receiving for the particular Slave station
D1002	1 ~ 2	Instruction code. 1: read data from Slave stations; 2: write data in Slave stations
D1003	1 ~ 64	Length of data transferred or received. (If the data designated is a 32-bit counter, the content value = 1 ~ 32)
D1004	1 ~ 6 10 ~ 13	Designates the device type of the Master station 1:Input Contact X 2:Output Contact Y 3:Auxiliary Coil M 4:State Coil S 5:Timer Contact T 6:Counter Contact C 10:The Present-value Register of the Timer 11:16-bit Counter, Present-value Register 12:32-bit Counter, Present-value Register 13:Data Register D
D1005		Designates the initial ID number of the Master station device
D1006	1 ~ 6 10 ~ 13	Designates the device type of the Slave station
D1007		Designates the initial ID number of the Slave station device
D1008	1 ~ 255	Designates the Slave station ID number
D1009	1 ~ 2	Instruction code
D1010	1 ~ 64	Length of data transferred/received
D1011	1 ~ 6 10 ~ 13	Designates the device type of the Master station
D1012		Designates the initial ID number of the Master station device
D1013	1 ~ 6 10 ~ 13	Designates the device type of the Slave station
D1014		Designates the initial ID number of the Slave station device
⋮	⋮	

Description of the 1st data transmission/receiving operation

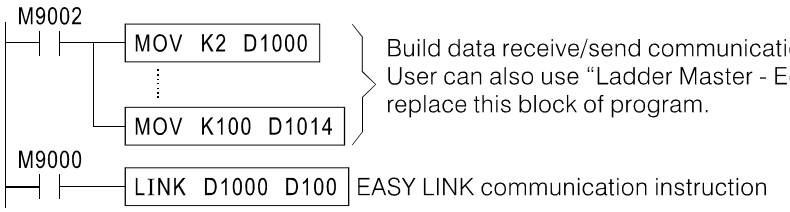
Description of the 2nd data transmission/receiving operation

- The attributes of the devices designated in a data transmission/receiving operation should be the same. For example, if the device designated by the Master station is a bit device, then the designated device of the Slave station should be also a bit device.

- The instruction working area headed with (S2):

(S2)		Description
D100	Lower 8 bits	The Slave station ID number when a communication error occurs
	Upper 8 bits	Instruction working status 0:Normal data transmission/receiving 2:Error of the length of the transferred/received data (unequal to 1 ~ 64) 4:Error of the designated device type 5:Error of the designated device ID number 6:The attributes of the designated devices by the Master and Slave stations are different A:Normal communications but no response from Slave stations B:Abnormal communications
D101 ~ D103	The working area required when the instruction is performed	

- Programming Example



There are totally 2 transmission/receiving data sets in this example.

① Read D10 ~ D19 of Slave station #5 to D0 ~ D9 of the Master station

② Write M0 ~ M29 of the Master station to M100 ~ M129 of Slave station #2.

ⓢ1	Content Value	
D1000	2	Two transmission/receiving data sets
D1001	5	Designates Slave station #5
D1002	1	Reads data from the Slave station
D1003	10	Length of the data to be read
D1004	13	Designates the device headed with the Master station as D0
D1005	0	
D1006	13	Designates the device headed with the Slave station as D10
D1007	10	
D1008	2	Designates Slave station #2
D1009	2	Write data to the Slave station
D1010	30	Length of the data to be written
D1011	3	Designates the device headed with the Master station as M0
D1012	0	
D1013	3	Designates the device headed with the Slave station as M100
D1014	100	

The 1st transmission/receiving data set:
D10 ~ D19 of Slave station #5
↓
D0 ~ D9 of the Master

The 2nd transmission/receiving data set:
M0 ~ M29 of the Master
↓
M100 ~ M129 of Slave station #2

- Edit Communication Table

Besides using program to build data receiving/sending communication table, Ladder Master provides a more user-friendly data input interface to let the users build communication table.
Select the Ladder Master "Tools ---- Edit Communication Table ...." menu to enter the communication table edition screen. Through a step-by-step guiding window, the user can easily create and edit communication table.
After the edition is done, the communication data will be stored into file register assigned by the user, and the table is created. This function also allows the user to retrieve the table data back from file register for editing.
For VB series PLCs, the file register is read-only, and its value will be treated as part of the user program. When user copy or save program file, the file register together with the program itself will be copied or saved. This feature makes the file register very suitable for communication table storing. It can be easily copied from and helps to save PLC program space. For detailed introduction on file register, please refer to "2-9 File Register (D)".

- Communication Table Example

Instruction: LINK ▼ Table Starting Position: D1000 Table Length: 15

Number	Command	Master Data		Slave ID	Slave Data	Length	Word / Bit
1	Read	D0	<--	5	D10	10	W
2	Write	M0	-->	2	M100	30	B

MEMO

MEMO

6-11 Serial Communication Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
110	D	ECMP	P	Compares two BIN floating point values		○	
111	D	EZCP	P	Compares a BIN float range with a BIN float value		○	
118	D	EBCD	P	Converts BIN floating point format to DEC format		○	
119	D	EBIN	P	Converts DEC format to BIN floating point format		○	
120	D	EADD	P	Adds up two BIN floating point numbers		○	
121	D	ESUB	P	Subtracts one BIN floating point number from another		○	
122	D	EMUL	P	Multiplies two BIN floating point numbers		○	
123	D	EDIV	P	Divides one BIN floating point number from another		○	
127	D	ESQR	P	Square root of a BIN floating point value		○	
129	D	INT	P	BIN floating point → BIN integer format		○	
130	D	SIN	P	Calculates the sine of a BIN floating point value		○	
131	D	COS	P	Calculates the cosine of a BIN floating point value		○	
132	D	TAN	P	Calculates the tangent of a BIN floating point value		○	

D	FNC 110 ECMP	P	DECMP P (S1) (S2) (D)	Compares two BIN floating point values	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○				○	○
S2											○				○	○
D		○	○	○												○
• D occupies 3 consecutive devices																

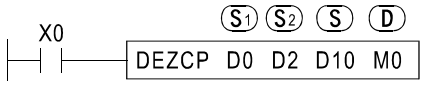
S1 : Comparative value data 1
S2 : Comparative value data 2
D : Comparison result

- The data of (S1) is compared to the data of (S2). The result is indicated by 3 bit devices which are specified with the head address entered as (D).
- When X0="ON", this instruction is activated.
  If (D1,D0) the double BIN floating number (S1) > (D11,D10) the double BIN floating number (S2), then M0="ON".
  If (D1,D0) the double BIN floating number (S1) = (D11,D10) the double BIN floating number (S2), then M1="ON".
  If (D1,D0) the double BIN floating number (S1) < (D11,D10) the double BIN floating number (S2), then M2="ON".
- When X0 turns "OFF", this instruction is deactivated. Then, the "ON"/"OFF" status of M0 ~ M2 will be kept the event before X0="OFF".
- This instruction is a 32 bit instruction, therefore must use DECMP or DECMPP in a program.
- Please combine two of M0 ~ M2when the result ≤ , ≥ or ≠ is needed.
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the comparison function.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.

D	FNC 111 EZCP	P	DEZCP P (S1) (S2) (S) (D)	Compares a BIN float range with a BIN float value	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○				○	○
S2											○				○	○
S											○				○	○
D		○	○	○												○

- D occupies 3 consecutive devices
- S1 ≤ S2



S1 : Upper limit of the data range
S2 : Lower limit of the data range
S : Comparative value
D : Compared result, occupies 3 consecutive devices

- The value of (S) is compared to the data range between (S1) and (S2). The result is indicated by 3 bit devices which are specified with the head address entered as (D).
- When X0="ON", this instruction is activated.

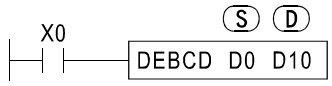
  If (D11,D10) the double BIN floating number (S) < (D1,D0) the double BIN floating number (S1), then M0="ON".

  If (D1,D0) the double BIN floating number (S1) <= (D11,D10) the double BIN floating number (S) <= (D2,D3) the double BIN floating number (S2), then M1="ON".

  If (D11,D10) the double BIN floating number (S) > (D3,D2) the double BIN floating number (S2), then M2="ON".
- When X0 turns "OFF", this instruction is deactivated. Then, the "ON"/"OFF" status of M0 ~ M2 will be kept the event before X0="OFF".
- This instruction is a 32 bit instruction, therefore must use DEZCP or DEZCPP in a program.
- When (S1) > (S2), the value of (S1) will become both Upper/Lower Limit to compares with (S).
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the comparison function.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.

					M	VB	VH
D	FNC 118 EBCD	P	DEBCDP (S) (D)	Converts BIN floating point format to DEC format		○	
D	FNC 119 EBIN	P	DEBINP (S) (D)	Converts DEC format to BIN floating point format		○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D											○					○



S : Source Device of Transfer

D : Destination Device

- When X0= "ON", this instruction is activated. It will uses the BIN format value in (D1,D0) to convert the number to a DEC format number then moves the value into (D11,D10).
- This instruction is a 32 bit instruction, therefore must use DEBCD or DEBCDP in a program.
- Ex. If the content value of (D1,D0) is 1.234 $\times 10^2$, then after the convert, (D10)=1234 and (D11)= -1.

S : Source Device of Transfer

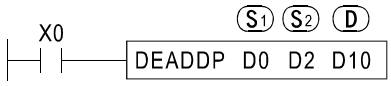
D : Destination Device

- When X1= "ON", this instruction is activated. It will uses the DEC format value in (D3,D2) to convert the number to a BIN format number then moves the value into (D13,D12).
- This instruction is a 32 bit instruction, therefore must use DEBIN or DEBINP in a program.
- Ex. If the content values of (D2) = 2345 and (D3)=5, then after the convert, the content value of (D13,D12) is 2.345$\times 10^8$.

- All of floating point number will occupy two Registers.
- Please refer to CH 2-12 "Numerical System", for the formats of BIN and DEC floating point numbers are stored in Registers.

D	FNC 120 EADD	P	DEADDP (S1) (S2) (D)	Adds up two BIN floating point numbers	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○				○	○
S2											○				○	○
D											○					○



S1 : Summand
S2 : Addend
D : Total

- When X0= "OFF" → "ON", the BIN floating point summand (D1,D0) will be added to the Bin floating point addend (D3,D2), and the total will be stored at the specified destination devices (D11,D10).

1.235×10^{2} (D1,D0) BIN floating point number
+ 3.2×10^{0} (D3,D2) BIN floating point number
1.267×10^{2} (D11,D10) BIN floating point number

- This instruction is a 32 bit instruction, therefore must use DEADD or DEADDP in a program.
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the addition function.
- To execute this instruction, the result will reacted on the status of flags.

  If the result of the calculation is equal to zero, the zero flag M9020="ON".

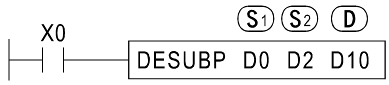
  If the value of calculated result (D) exceeds the available range of a BIN floating point number (including positive and negative), then the carry flag M9021= "ON" and the result (D) is set to the largest value.

  If the value of calculated result (D) is smaller than the available range of a BIN floating point number (including positive and negative), then the borrow flag M9022= "ON" and the result (D) is set to the smallest value.

  The available value range of a BIN floating point number, please refer to CH 2-12 "Numerical System"
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.

D	FNC 121 ESUB	P	DESUBP S1 S2 D	Subtracts one BIN floating point number from another	M	VB	VH
						○	

Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○				○	○
S2											○				○	○
D											○					○



S1 : Minuend

S2 : Subtrahend

D : Remainder

- When X0= "OFF" → "ON", the BIN floating point subtrahend (D3,D2) will be subtracted from the BIN floating point minuend (D1,D0), and the remainder will be stored at the specified destination devices (D11,D10).

	1.235×10^2	(D1,D0) BIN floating point number
−	3.2×10^0	(D3,D2) BIN floating point number
	1.203×10^2	(D11,D10) BIN floating point number

- This instruction is a 32 bit instruction, therefore must use DESUB or DESUBP in a program.
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the subtraction function.
- To execute this instruction, the result will reacted on the status of flags.

  If the result of the calculation is equal to zero, the zero flag M9020="ON".

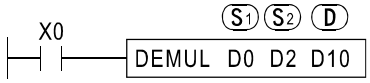
  If the value of calculated result (D) exceeds the available range of a BIN floating point number (including positive and negative), then the carry flag M9021= "ON" and the result (D) is set to the largest value.

  If the value of calculated result (D) is smaller than the available range of a BIN floating point number (including positive and negative), then the borrow flag M9022= "ON" and the result (D) is set to the smallest value.

  The available value range of a BIN floating point number, please refer to CH 2-12 "Numerical System"
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.

D	FNC 122 EMUL	P	DEMULP (S1) (S2) (D)	Multiplies two BIN floating point numbers	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○				○	○
S2											○				○	○
D											○					○



S1 : Minuend
S2 : Subtrahend
D : Remainder

- When X0= "ON", the BIN floating point multiplicand (D1,D0) will be multiplied by the BIN floating point multiplier (D3,D2), and the product will be stored at the specified destination devices (D11,D10).

3.14×10^{0} (D1,D0) BIN floating point number
× 2.3×10^{1} (D3,D2) BIN floating point number
7.222×10^{1} (D11,D10) BIN floating point number

- This instruction is a 32 bit instruction, therefore must use DEMUL or DEMULP in a program.
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the multiply function.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.

D	FNC 123 EDIV	P	DEDIVP (S1) (S2) (D)	Divides one BIN floating point number from another	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○				○	○
S2											○				○	○
D											○					○

S1 : Dividend
S2 : Divisor
D : Quotient

- When X0= “OFF” → “ON”, the BIN floating point dividend (D1,D0) will be divided by the BIN floating point divisor (D3,D2), and the quotient will be stored at the specified destination devices (D11,D10).

1.23×10^{4} (D1,D0) BIN floating point number

÷ 3.0×10^{-1} (D3,D2) BIN floating point number

4.1×10^{4} (D11,D10) BIN floating point number

- This instruction is a 32 bit instruction, therefore must use DEDIV or DEDIVP in a program.
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the division function.
- All of floating point number will occupy two Registers, please refer to CH 2-12 “Numerical System” for the format of a floating point number is stored in Registers.
- PLC will identify an operation error, if the divisor (S2) is equal to “0”.

D	FNC 127 ESQR	P	DESQRP (S) (D)	Square root of a BIN floating point value	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○				○	○
D											○					○

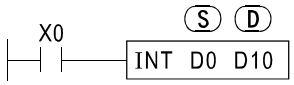
S : Source device

D : Storage device for the result of square root

- This square root function is performed on the specified BIN floating point value of (S) and the result is stored on (D).
- When X0= "ON", the function is activated, uses the BIN floating point source (D1,D0) to get its square root, and the result will be stored at the specified destination devices (D11,D10) by BIN floating point format.
- This instruction is a 32 bit instruction, therefore must use DESQR or DESQRP in a program.
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the square root function.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.
- If the result of the calculation is equal to zero, the zero flag M9020= "ON".
- (S) can be assigned to a positive number only, if (S) is a negative then an error occurs and the error flag M9067="ON".

D	FNC 129 INT	P	DINTP (S) (D)	BIN floating point BIN integer format	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D											○					○



S : Source device

D : Converted result

- When X0= “ON”, the function is activated, uses the BIN floating point source (D1,D0) to convert the value to a equal or nearest smaller BIN integer format number, the result will be stored at the specified destination device (D10) and the number behind decimal point will be rejected.
- If the result of the conversion is equal to zero, the zero flag M9020= “ON”.

  If the number behind decimal point has been rejected, the borrow flag M9021= “ON”.

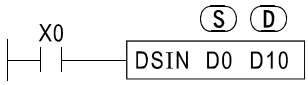
  If the result is exceed the range below, the borrow flag M9022= “ON” to indicate overflow.

  16 bit instruction: – 32,768 ~ 32,767

  32 bit instruction: – 2,147,483,648 ~ 2,147,483,647
- All of floating point number will occupy two Registers, please refer to CH 2-12 “Numerical System” for the format of a floating point number is stored in Registers.

D	FNC 130 SIN	P	DSINP (S) (D)	Calculates the sine of a BIN floating point value	M	VB	VH
						○	

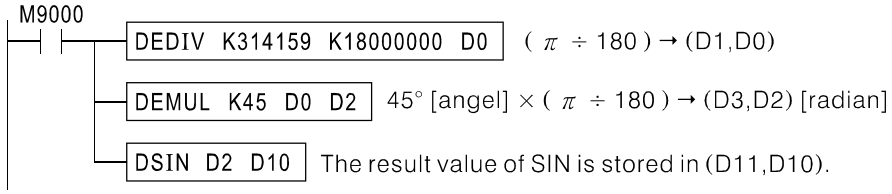
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D											○					○



S : Source device for the radians angle

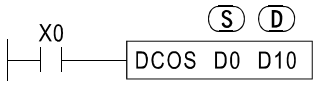
D : Calculated result

- This instruction preforms the mathematical SIN operation on the floating point value in (S) (radian), the result is stored in (D) .
- When X0= "ON", the function is activated, uses the BIN floating point radian (D1,D0) to calculate the sine value and the result will be stored at the specified destination devices (D11,D10).
- Radian = Degree × π ÷180
- This instruction is a 32 bit instruction, therefore must use DSIN or DSINP in a program.
- In this instruction, both (S) and (D) are BIN floating point numbers.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.
- Below is an program example of how to calculate angles (45°) in radian using floating point, then use the radian to get the value of sine.



D	FNC 131 COS	P	DCOSP (S) (D)	Calculates the cosine of a BIN floating point value	M	VB	VH
						○	

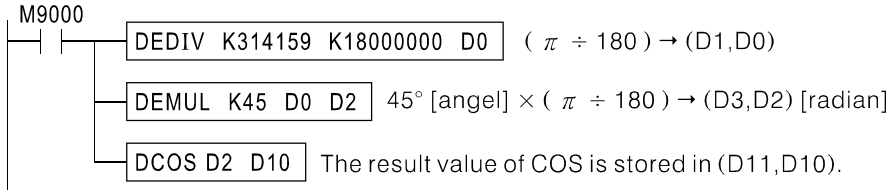
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D											○					○



S : Source device for the radians angle

D : Calculated result

- This instruction preforms the mathematical COS operation on the floating point value in (S) (radian), the result is stored in (D).
- When X0= "ON", the function is activated, uses the BIN floating point radian (D1,D0) to calculate the cosine value and the result will be stored at the specified destination devices (D11,D10).
- Radian = Degree × π ÷180
- This instruction is a 32 bit instruction, therefore must use DCOS or DCOSP in a program.
- In this instruction, both (S) and (D) are BIN floating point number.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.
- Below is an program example of how to calculate angles (45°) in radian using floating point, then use the radian to get the value of cosine.



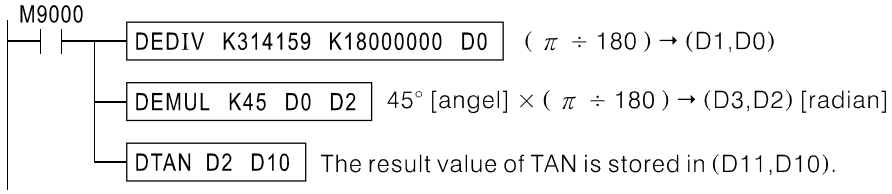
D	FNC 132 TAN	P	DTANP (S) (D)	Calculates the tangent of a BIN floating point value	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D											○					○

S : Source device for the radians angle

D : Calculated result

- This instruction preforms the mathematical TAN operation on the floating point value in (S) (radian) , the result is stored in (D) .
- When X0= “ON”, the function is activated, uses the BIN floating point radian (D1,D0) to calculate the tangent value and the result will be stored at the specified destination devices (D11,D10).
- Radian = Degree × π ÷180
- This instruction is a 32 bit instruction, therefore must use DTAN or DTANP in a program.
- In this instruction, both (S) and (D) are BIN floating point number.
- All of floating point number will occupy two Registers, please refer to CH 2-12 “Numerical System” for the format of a floating point number is stored in Registers.
- Below is an program example of how to calculate angles (45°) in radian using floating point, then use the radian to get the value of tangent.



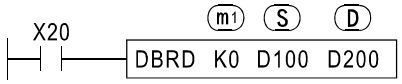
MEMO

6-12 Others

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
90		DBRD	P	Reads data from the data bank	○	○	
91		DBWR	P	Writes data into the data bank	○	○	
147	D	SWAP	P	Swaps high/low byte	○	○	
176		TFT		Reads data from the data bank	○	○	○
177		TFH		Reads data from the data bank	○	○	○
178		TFK		Reads data from the data bank	○	○	○

FNC 90 DBRD	P	DBRDP (m1) (S) (D)	Reads data from the data bank	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
m1															○	
S											○	○			○	○
D											○	○				○
• m1=0 • M series: S = 0 ~ 1021 • VB series: S = 0 ~ 2045																



m1: The location of data bank

S : The data in specific page of data bank will be read

D : The initial ID of specified registers, which are assigned as the data storage

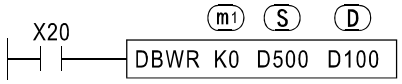
- The M and VB series PLC are able to install a data bank, it can store and apply huge data.

Data Bank	M series	VB series
Model number	M-DB1	VB-DB1R
Component parts	Flash ROM	SRAM + Lithium battery
Memory capacity	1022 pages (64 Words / page)	2046 pages (64 Words / page)

- The M series PLC can use this instruction to read the data in the M-DB1 data bank.
- The VB series PLC can use this instruction to read the data in the VB-DB1R data bank.
- If D100=3 and X20= “ON”, it will execute to read the data in page 3 of the data bank and put the data in D200 ~ D263.
- One page of a data bank can store 64 registers’ data.
- When X20= “OFF”, the instruction will not be performed but the data (which was read previously) will still remain.

FNC 91 DBWR	P	DBWRP (m1) (S) (D)	Writes data into the data bank	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
m1															○	
S											○	○				○
D											○	○			○	○
• m1=0 • M series: D = 0 ~ 1021 • VB series: D = 0 ~ 2045																



m1: The location of data bank

S : The source data in specific registers (which are starting from S), would be written into the data bank

D : The specific page in data bank will be covered

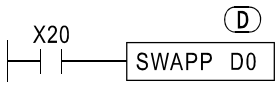
- The M and VB series PLC are able to install a data bank, it can store and apply huge data.

Data Bank	M series	VB series
Model number	M-DB1	VB-DB1R
Component parts	Flash ROM	SRAM + Lithium battery
Memory capacity	1022 pages (64 Words / page)	2046 pages (64 Words / page)

- The M series PLC can use this instruction to write the data into the M-DB1 data bank.
- The VB series PLC can use this instruction to write the data into the VB-DB1R data bank.
- If D100=4 and X20= “ON”, it will read the data from registers D500 ~ D563 and write the data into page 4 of the data bank.
- One page of a data bank can store 64 registers’ data.
- Since the M-DB1 is using the Flash ROM technique to storage data. Even though, in every page of the memory, the rewrite operate is available to be used more than 10,000 times. But, it still has the limit. So, when the program using the instruction DBWR to rewrite data into M-DB1, better change it to the instruction DBWRP. The DBWRP can avoid useless operate of rewrite, and then extend the lifespan of the Flash ROM. The VB series rewrite operate times is unlimited.
- When M series CPU module rewrites data to a M-DB1, every single page needs 10ms to execute the function. And at the same time, other executing function will be interrupted. The current value of Watch Dog timer will be reset. The VB series won't has this reaction.

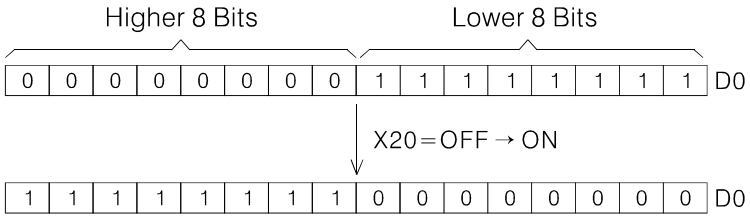
D	FNC 147 SWAP	P	DSWAPP (D)	Byte Swap	M	VB	VH
					○	○	

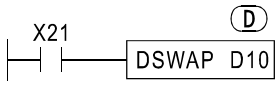
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D						○	○	○	○	○	○	○		○		○



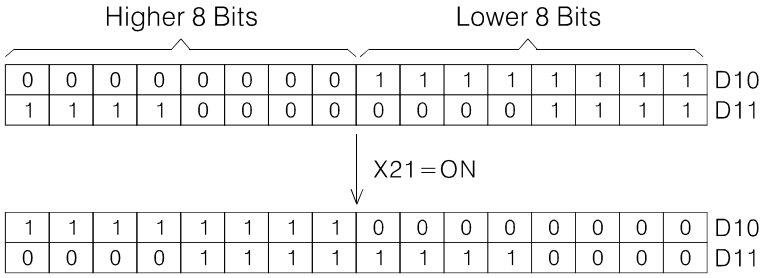
D : Device which Higher/Lower 8 bits are to be exchanged

- When X20="OFF" → "ON", Higher 8 bits and Lower 8 bits of (D0) will be exchanged.



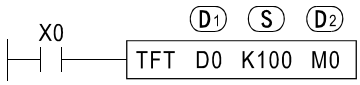


- When X21= "ON", Higher 8 bits and Lower 8 bits of (D10) will be exchanged. And also, Higher 8 bits and Lower 8 bits of (D11) will be exchanged.



FNC 176 TFT	TFT (D1) (S) (D2)	Timer (10 ms)	M	VB	VH
			○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D1											○	○				○
S											○	○			○	○
D2		○	○	○												○
• S=0～32767, otherwise S=0																

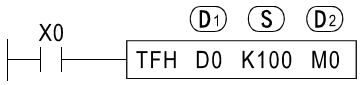


D1 : The current value of the timer (unit=10ms.)
S : The setting value of the timer (unit=10ms.)
D2 : The output contact of the timer

- The unit of this instruction is used the 10ms. base timer.
- The timer count the time by up counting clock pulses. When the Current value (D1) = Setting value (S) (the value designated to a Timer), then the Timer's contact (D2) will be activated (ON).
- This timer's real setting value = 10 ms. × setting value (S) .
- The example above:
  When X0= "ON", the current value of the timer starts to count clock pulses (by unit: 10 ms). When the current value reaches the setting value K100 (1 second), the contact M0= "ON".
  When input contact X0 = "OFF" or the power failure, the Current value of Timer will return to "0" and the contact will become "OFF".

FNC 177 TFH	TFH (D1) (S) (D2)	Timer (100 ms)	M	VB	VH
			○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D1											○	○				○
S											○	○			○	○
D2		○	○	○												○
• S=0～32767, otherwise S=0																

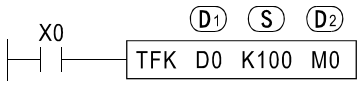


D1 : The current value of the timer (unit=100ms.)
S : The setting value of the timer (unit=100ms.)
D2 : The output contact of the timer

- The unit of this instruction is used the 100ms. base timer.
- The timer count the time by up counting clock pulses. When the Current value (D1) = Setting value (S) (the value designated to a Timer), then the Timer's contact (D2) will be activated (ON).
- This timer's real setting value = 100 ms. × setting value (S) .
- The example above:
  When X0= "ON", the current value of the timer starts to count clock pulses (by unit: 100 ms). When the current value reaches the setting value K100 (10 second), the contact M0= "ON".
  When input contact X0 = "OFF" or the power failure, the Current value of Timer will return to "0" and the contact will become "OFF".

FNC 178 TFK	TFK (D1) (S) (D2)	Timer (1 second)	M	VB	VH
			○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D1											○	○				○
S											○	○			○	○
D2		○	○	○												○
• S=0～32767, otherwise S=0																



D1 : The current value of the timer (unit=1 sec.)
S : The setting value of the timer (unit=1 sec.)
D2 : The output contact of the timer

- The unit of this instruction is used the 1 sec. base timer.
- The timer count the time by up counting clock pulses. When the Current value (D1) = Setting value (S) (the value designated to a Timer), then the Timer's contact (D2) will be activated (ON).
- his timer's real setting value = 1 sec. × setting value (S) .
- The example above:
  When X0= “ON”, the current value of the timer starts to count clock pulses (by unit: 1 second). When the current value reaches the setting value K100 (100 second), the contact M0= “ON”.
  When input contact X0 = “OFF” or the power failure, the Current value of Timer will return to “0” and the contact will become “OFF”.

MEMO

6-13 Serial Communication Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
160		TCMP	P	Time Compare	○	○	
161		TZCP	P	Time Zone Compare	○	○	
162		TADD	P	Time Add	○	○	
163		TSUB	P	Time Subtract	○	○	
166		TRD	P	Read RTC data	○	○	
167		TWR	P	Set RTC data	○	○	○
170	D	GRY	P	Converts BIN to Gray code	○	○	
171	D	GBIN	P	Converts Gray code to BIN	○	○	

FNC 160 TCMP	P	TCMP P (S1) (S2) (S3) (S) (D)	Time Compare	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
S_3					○	○	○	○	○	○	○	○		○	○	○
S									○	○	○					○
D		○	○	○												○

- S_1=0 ~ 23; S_2=0 ~ 59; S_3=0 ~ 59
- Both S and D occupy 3 consecutive devices respectively

S_1 : The "Hour" value of the time set
S_2 : The "Minute" value of the time set
S_3 : The "Second" value of the time set
S : Time compare value
D : The storages of compare result

- Compare the setting values (Hours, Minutes and Seconds which are designated by (S1) ~ (S3)) to the time value (specified by the head ID (S) of 3 consecutive data devices), and the result of Comparison is stored to (D).
- When X20= "ON", the instruction will be performed.

If 8: 30'20" > | D0 (Hour) / D1 (Minute) / D2 (Second) | then M0= "ON".

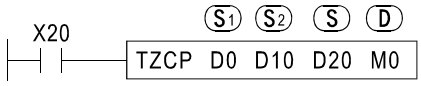
If 8: 30'20" = | D0 (Hour) / D1 (Minute) / D2 (Second) | then M1= "ON".

If 8: 30'20" < | D0 (Hour) / D1 (Minute) / D2 (Second) | then M2= "ON".

- The current time of the real time clock is stored in Special Registers D9013 ~ D9015. D9015 (Hour), D9014 (Minute), D9013 (Second)
- When X20= "OFF", the instruction will not be performed. M0 ~ M2 will remain the status before X20= "OFF".
- Please combine two of M0 ~ M2when the result ≤ , ≥ or ≠ is needed.
- If the content value of the register designated by (S) exceeding the time value required, it will be regarded as an operation error.

FNC 161 TZCP	P	TZCP(P) (S1) (S2) (S) (D)	Time Zone Compare	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1									○	○	○					○
S2									○	○	○					○
S									○	○	○					○
• D		○	○	○												○
• S1 ≤ S2	• All S1, S2, S and D will occupy 3 consecutive devices respectively															



S1 : Lower limit of the setting time period
S2 : Upper limit of the setting time period
S : Time compare value
D : The storages of compare result

- The time compare value is defined by (S), it will be compared to the lower limit of the setting time period defined by (S1) and the upper limit of the setting time period defined by (S2). And then, the compare result will be stored in (D).
- When X20= "ON", the instruction will be performed.

(S1)		(S)	
D0 (Hour)	>	D20 (Hour)	,M0= "ON"
D1 (Minute)		D21 (Minute)	
D2 (Second)		D22 (Second)	

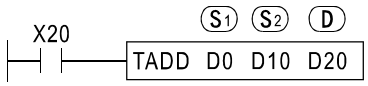
		(S)		(S2)	
D0 (Hour)	≤	D20 (Hour)	≤	D10 (Hour)	,M1= "ON"
D1 (Minute)		D21 (Minute)		D11 (Minute)	
D2 (Second)		D22 (Second)		D12 (Second)	

D20 (Hour)	>	D10 (Hour)	,M2= "ON"
D21 (Minute)		D11 (Minute)	
D22 (Second)		D12 (Second)	

- The current time of the real time clock is stored in Special Registers D9013 ~ D9015. D9015 (Hour), D9014 (Minute), D9013 (Second)
- When X20= "OFF", the instruction will not be performed. M0 ~ M2 will remain the status before X20= "OFF".
- When (S1) > (S2), the value of (S1) will become both Upper/Lower Limit to compares with (S).
- If the content value of the register designated by (S1), (S2) or (S1) exceeding the time value required, it will be regarded as an operation error.

FNC 162 TADD	P	TADDP (S1) (S2) (D)	Time Addition	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1									○	○	○					○
S2									○	○	○					○
D									○	○	○					○
• All S1, S2, S and D will occupy 3 consecutive devices respectively																



S1 : Summand of the time value
S2 : Addend of the time value
D : The addition result

- The time value defined by (S1) is added to the time value defined by (S2) and the result is stored in the registers defined by (D).
- When X20= “ON”, the time addition is performed.

(S1)		(S2)		(D)
D0 8(Hour)		D10 6(Hour)		D20 15(Hour)
D1 30(Minute)	+	D11 35(Minute)	=	D21 5(Minute)
D2 0(Second)		D12 30(Second)		D22 30(Second)
8:30:0		6:35:30		15:5:30

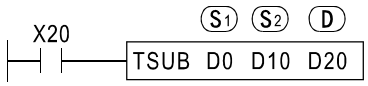
- If the result of the time addition is longer than 24 hours, then the Carry Flag M9022 will be set ”ON” and (D) will display the value where 24 hours is subtracted from the total.

(S1)		(S2)		(D)	
D0 8(Hour)		D10 20(Hour)		D20 4(Hour)	
D1 25(Minute)	+	D11 10(Minute)	=	D21 35(Minute)	,And M9022= “ON”
D2 30(Second)		D12 20(Second)		D22 50(Second)	
8:25:30		20:10:20		4:35:50	
				28-24=4	

- If the result of the time addition equals “0” (0 hour 0 min 0 sec), then the Zero Flag M9020 will be set “ON”.
- If the content value of the register designated by (S1) or (S2) exceeding the time value required, it will be regarded as an operation error.

FNC 163 TSUB	P	TSUBP (S1) (S2) (D)	Time Subtraction	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S_1									○	○	○					○
S_2									○	○	○					○
D									○	○	○					○
• All S_1, S_2, S and D will occupy 3 consecutive devices respectively																



S_1 : Minuend of the time value
S_2 : Subtrahend of the time value
D : The subtraction result

- The time value defined by (S1) is subtracted by the time value defined by (S2) and the result is stored in the register defined by (D).
- When X20 = "ON", the time subtraction is performed.

(S1)		(S2)		(D)
D0 18 (Hour)		D10 8 (Hour)		D20 9 (Hour)
D1 28 (Minute)	−	D11 40 (Minute)	=	D21 48 (Minute)
D2 50 (Second)		D12 20 (Second)		D22 30 (Second)
18:28:50		8:40:20		9:48:30

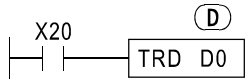
- If the result of the time subtraction is a negative, then the Borrow Flag M9021 will be set "ON" and (D) will display the value where the negative value is added to 24 hours.

(S1)		(S2)		(D)	
D0 6 (Hour)		D10 20 (Hour)		D20 10 (Hour)	
D1 30 (Minute)	−	D11 20 (Minute)	=	D21 10 (Minute)	,And M9021 = "ON"
D2 20 (Second)		D12 10 (Second)		D22 10 (Second)	
6:30:20		20:20:10		10:10:10	
				↑ (-4)+24=10	

- If the result of the time subtraction equals "0" (0 hour 0 min 0 sec), then Zero Flag M9020 will be set "ON".
- If the content value of the register designated by (S1) or (S2) exceeding the time value required, it will be regarded as an operation error.

FNC 166 TRD	P	TRDP (D)	Time Read	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
D									○	○	○					○
• D will occupy 7 consecutive devices respectively																



D : The subtraction result

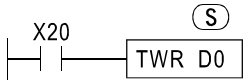
- M Series PLC's M1-CPU1 can install the M-RTC, M-232R or M-485R expansion card. After one of those expansion card has been installed, the PLC will be provide with the real time clock functions. The real time clock has seven sets of data such as year, month, day, hour, minute, second and week, the data will be stored in Special Register D9013 ~ D9019.
- VB Series PLC's Main Unit can install the VB-RTC, VB-MP1R or VB-DB1R expansion card. After one of those expansion card has been installed, the PLC will be provide with the real time clock functions. The real time clock has seven sets of data such as year, month, day, hour, minute, second and week, the data will be stored in Special Register D9013 ~ D9019.
- Programmers do not need to memorize the location of real time clock is stored, they can use this instruction to read the current time and date of the real time clock and store the data to contiguous 7 registers which is specified by (D).
- When X20= "ON", as the diagram below, the data of the real time clock will be read and stored into designated registers D0 ~ D6.

Item	Special Register ID	Content Value of the RTC		
Year	D9018	2000 ~ 2099	Read and Store	D0
Month	D9017	1 ~ 12		D1
Day	D9016	1 ~ 31		D2
Hour	D9015	0 ~ 23		D3
Minute	D9014	0 ~ 59		D4
Second	D9013	0 ~ 59		D5
Week	D9019	0 ~ 6		D6

- The content value of D9019=0 represents Sunday
  The content value of D9019=1 represents Monday
  The content value of D9019=2 represents Tuesday
  The content value of D9019=3 represents Wednesday
  The content value of D9019=4 represents Thursday
  The content value of D9019=5 represents Friday
  The content value of D9019=6 represents Saturday

FNC 167 TWR	P	TWRP (S)	Time Write	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S									○	○	○					○



S : The source registers which store the new current value of the real time clock

- M Series PLC's M1-CPU1 can install the M-RTC, M-232R or M-485R expansion card. After one of those expansion card has been installed, the PLC will be provide with the real time clock functions. The real time clock has seven sets of data such as year, month, day, hour, minute, second and week, the data will be stored in Special Register D9013 ~ D9019.
- VB Series PLC's Main Unit can install the VB-RTC, VB-MP1R or VB-DB1R expansion card. After one of those expansion card has been installed, the PLC will be provide with the real time clock functions. The real time clock has seven sets of data such as year, month, day, hour, minute, second and week, the data will be stored in Special Register D9013 ~ D9019.
- When X20= "ON", as the diagram below, the data in designated source registers D0 ~ D6 will be read and reset the current value of real time clock.

(S)	Content Value of the RTC		Special Register ID		Content Value of the Real Time Clock
D0	2000 ~ 2099		D9018		Year
D1	1 ~ 12		D9017		Month
D2	1 ~ 31	Rewrite ⇨	D9016	Rewrite ⇨	Day
D3	0 ~ 23		D9015		Hour
D4	0 ~ 59		D9014		Minute
D5	0 ~ 59		D9013		Second
D6	0 ~ 6		D9019		Week

- The content value (0 ~ 6) of D6 represents Sunday, Monday…Saturday.
- The content value of the source registers (defined by (S)) exceeding the valid range (as shown above), it will be regarded as an operation error.
- Also can use the program develop software Ladder Master to perform setting of the real time clock (rewrite RTC data).

D	FNC 170 GRY	P	DGRYP (S) (D)	Converts BIN to Gray Code	M	VB	VH
					○	○	

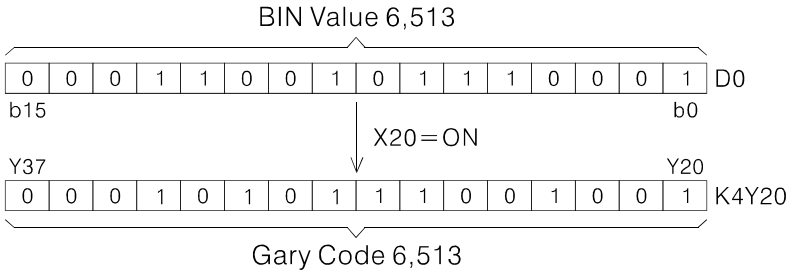
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

- For a 16-bit instruction, S= 0 ~ 32767
- For a 32-bit instruction, S= 0 ~ 2147483647

S : Source device (Gary Code)

D : The destination device where the converted BIN value is stored

- When the instruction is performed, it converts the content BIN value of the source devices (designated by (S)) into Gary Code and transfers they to the designated devices (D).
- When X20= "ON", the content value of (D0) will be converted to Gary Code and then 16 output points (Y0 ~ Y37) will be exported to the terminals.



- The valid range of (S) is shown below. Any value exceeding such a range will be regarded as an operation error.
  For a 16-bit instruction: 0 ~ 32,767
  For a 32-bit instruction: 0 ~ 2,147,483,647

D	FNC 171 GBIN	P	DGBINP (S) (D)	Converts Gray Code to BIN	M	VB	VH
					○	○	

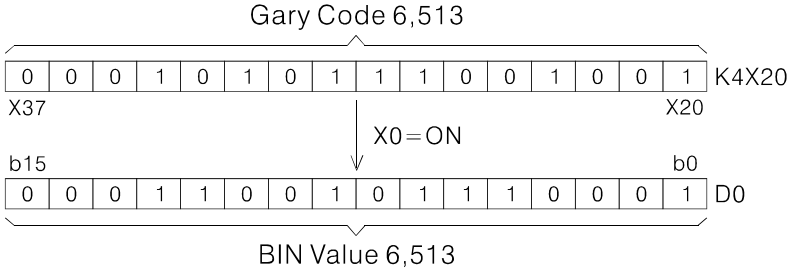
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

- For a 16-bit instruction, S= 0 ~ 32767
- For a 32-bit instruction, S= 0 ~ 2147483647

S : Source device where the Gary Code is stored

D : The destination device where the converted BIN value is stored

- When the instruction is performed, it converts the content Gary Code of the source devices (designated by (S)) into BIN Value and transfers they to the designated device (D).
- This instruction is always used to convert the code from an Absolute Rotary Encoder (which is connected to the PLC's input terminal and generally uses the Gary Code) to a BIN Value and transfer it to the register in the PLC.
- When X0= "ON", the code of an Absolute Rotary Encoder connected to 16 output points (Y20 ~ Y37) will be converted to BIN Value and then transfered to D0.



- The valid range of (S) is shown below. Any value exceeding such a range will be regarded as an operation error.
  For a 16-bit instruction: 0 ~ 32,767
  For a 32-bit instruction: 0 ~ 2,147,483,647

MEMO

6-14 In-line Comparisons

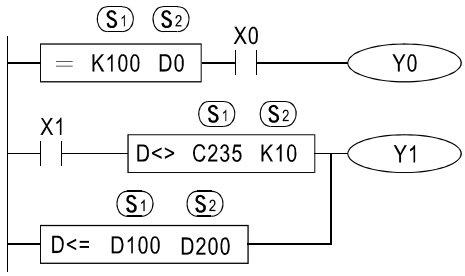
FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
224	D	LD=		Initial comparison contact. Active when (S1)=(S2)		○	○
225	D	LD>		Initial comparison contact. Active when (S1)>(S2)		○	○
226	D	LD<		Initial comparison contact. Active when (S1)<(S2)		○	○
228	D	LD<>		Initial comparison contact. Active when (S1)≠(S2)		○	○
229	D	LD<=		Initial comparison contact. Active when (S1)≤(S2)		○	○
230	D	LD>=		Initial comparison contact. Active when (S1)≥(S2)		○	○
232	D	AND=		Serial comparison contact. Active when (S1)=(S2)		○	○
233	D	AND>		Serial comparison contact. Active when (S1)>(S2)		○	○
234	D	AND<		Serial comparison contact. Active when (S1)<(S2)		○	○
236	D	AND<>		Serial comparison contact. Active when (S1)≠(S2)		○	○
237	D	AND<=		Serial comparison contact. Active when (S1)≤(S2)		○	○
238	D	AND>=		Serial comparison contact. Active when (S1)≥(S2)		○	○
240	D	OR=		Parallel comparison contact. Active when (S1)=(S2)		○	○
241	D	OR>		Parallel comparison contact. Active when (S1)>(S2)		○	○
242	D	OR<		Parallel comparison contact. Active when (S1)<(S2)		○	○
244	D	OR<>		Parallel comparison contact. Active when (S1)≠(S2)		○	○
245	D	OR<=		Parallel comparison contact. Active when (S1)≤(S2)		○	○
246	D	OR>=		Parallel comparison contact. Active when (S1)≥(S2)		○	○

					M	VB	VH
D	FNC 224 LD=		D= S1 S2	Initial comparison contact. Active when (S1)=(S2)		○	○
D	FNC 225 LD>		D> S1 S2	Initial comparison contact. Active when (S1)>(S2)		○	○
D	FNC 226 LD<		D< S1 S2	Initial comparison contact. Active when (S1)<(S2)		○	○
D	FNC 228 LD<>		D<> S1 S2	Initial comparison contact. Active when (S1)≠(S2)		○	○
D	FNC 229 LD<=		D<= S1 S2	Initial comparison contact. Active when (S1)≤(S2)		○	○
D	FNC 230 LD>=		D>= S1 S2	Initial comparison contact. Active when (S1)≥(S2)		○	○
D	FNC 232 AND=		D= S1 S2	Serial comparison contact. Active when (S1)=(S2)		○	○
D	FNC 233 AND>		D> S1 S2	Serial comparison contact. Active when (S1)>(S2)		○	○
D	FNC 234 AND<		D< S1 S2	Serial comparison contact. Active when (S1)<(S2)		○	○
D	FNC 236 AND<>		D<> S1 S2	Serial comparison contact. Active when (S1)≠(S2)		○	○
D	FNC 237 AND<=		D<= S1 S2	Serial comparison contact. Active when (S1)≤(S2)		○	○
D	FNC 238 AND>=		D>= S1 S2	Serial comparison contact. Active when (S1)≥(S2)		○	○
D	FNC 240 OR=		D= S1 S2	Parallel comparison contact. Active when (S1)=(S2)		○	○
D	FNC 241 OR>		D> S1 S2	Parallel comparison contact. Active when (S1)>(S2)		○	○
D	FNC 242 OR<		D< S1 S2	Parallel comparison contact. Active when (S1)<(S2)		○	○
D	FNC 244 OR<>		D<> S1 S2	Parallel comparison contact. Active when (S1)≠(S2)		○	○
D	FNC 245 OR<=		D<= S1 S2	Parallel comparison contact. Active when (S1)≤(S2)		○	○
D	FNC 246 OR>=		D>= S1 S2	Parallel comparison contact. Active when (S1)≥(S2)		○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○

S1 : The first source value of the comparison

S2 : The second source value of the comparison



When the content value of D0=K100 and X0= “ON”, then Y0= “ON”.

When X1= “ON” and the content value of C235≠K10, then Y1= “ON”.

OR, the content value of (D101, D100) is≤the content value of (D201,D200), then Y1= “ON”.

MEMO

MEMO