

搬移指令

FUN50 BDIST	位元組資料分配 ( BYTE DISTRIBUTE )	FUN50 BDIST																																																						
<div><div>執行控制-EN↑</div><div>50P.BDIST</div><div>S : N : D :</div></div> <div>S : 欲作位元組(Byte)分配之來源暫存器起始號碼 N : 欲分配之資料個數，單位為 Byte D : 存放分配資料之起始暫存器號碼 S，N，D 運算元可結合 V，Z 指標作間接定址應用。</div> <table><tr><th>運算元 \ 範圍</th><th>HR R0   R3839</th><th>ROR R5000   R8071</th><th>DR D0   D3071</th><th>K</th></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td></td></tr><tr><td>N</td><td>○</td><td>○</td><td>○</td><td>1~256</td></tr><tr><td>D</td><td>○</td><td>○*</td><td>○</td><td></td></tr></table>			運算元 \ 範圍	HR R0   R3839	ROR R5000   R8071	DR D0   D3071	K	S	○	○	○		N	○	○	○	1~256	D	○	○*	○																																			
運算元 \ 範圍	HR R0   R3839	ROR R5000   R8071	DR D0   D3071	K																																																				
S	○	○	○																																																					
N	○	○	○	1~256																																																				
D	○	○*	○																																																					
<div><div>● 當執行控制 “EN” =1 或 “EN ↑ ” ( P 指令 ) 由 0→1 時，將以 S 為起始之 N 個資料暫存器作位元組資料分配，並將資料分配結果存放到以 D 為起始之暫存器群。</div><div>● 當分配之資料個數不正確時，本指令不執行。</div><div>● PLC 與智慧型週邊透過通訊介面來作連結整合時，如果通訊間之資料型式為二進制而非 ASCII 碼方式時，需將 16 位元(Word)資料分配成 8 位元(Byte)資料後才能正確傳送出資料，本指令即可有效作此應用。</div></div> <div>程式範例：</div> <div><div>M2</div><div>EN</div><div>50P.BDIST</div><div>S : R 1000 N : R 999 D : R 1500</div></div> <div><div>範例說明：當 M2=1 時，以暫存器 R1000 為起始，暫存器 R999 之值為長度，作位元組分配，並將結果存放至暫存器 R1500 為起始之暫存器群。</div><div>本範例假設 R999=9，則存放位元組分配結果之暫存器為 R1500～R1508。</div><div><table><tr><th colspan="3">S</th></tr><tr><th></th><th>High Byte</th><th>Low Byte</th></tr><tr><td>R1000</td><td>Byte-0</td><td>Byte-1</td></tr><tr><td>R1001</td><td>Byte-2</td><td>Byte-3</td></tr><tr><td>R1002</td><td>Byte-4</td><td>Byte-5</td></tr><tr><td>R1003</td><td>Byte-6</td><td>Byte-7</td></tr><tr><td>R1004</td><td>Byte-8</td><td>Don't care</td></tr></table><table><tr><th colspan="3">D</th></tr><tr><th></th><th>High Byte</th><th>Low Byte</th></tr><tr><td>R1500</td><td>00</td><td>Byte-0</td></tr><tr><td>R1501</td><td>00</td><td>Byte-1</td></tr><tr><td>R1502</td><td>00</td><td>Byte-2</td></tr><tr><td>R1503</td><td>00</td><td>Byte-3</td></tr><tr><td>R1504</td><td>00</td><td>Byte-4</td></tr><tr><td>R1505</td><td>00</td><td>Byte-5</td></tr><tr><td>R1506</td><td>00</td><td>Byte-6</td></tr><tr><td>R1507</td><td>00</td><td>Byte-7</td></tr><tr><td>R1508</td><td>00</td><td>Byte-8</td></tr></table></div></div>			S				High Byte	Low Byte	R1000	Byte-0	Byte-1	R1001	Byte-2	Byte-3	R1002	Byte-4	Byte-5	R1003	Byte-6	Byte-7	R1004	Byte-8	Don't care	D				High Byte	Low Byte	R1500	00	Byte-0	R1501	00	Byte-1	R1502	00	Byte-2	R1503	00	Byte-3	R1504	00	Byte-4	R1505	00	Byte-5	R1506	00	Byte-6	R1507	00	Byte-7	R1508	00	Byte-8
S																																																								
	High Byte	Low Byte																																																						
R1000	Byte-0	Byte-1																																																						
R1001	Byte-2	Byte-3																																																						
R1002	Byte-4	Byte-5																																																						
R1003	Byte-6	Byte-7																																																						
R1004	Byte-8	Don't care																																																						
D																																																								
	High Byte	Low Byte																																																						
R1500	00	Byte-0																																																						
R1501	00	Byte-1																																																						
R1502	00	Byte-2																																																						
R1503	00	Byte-3																																																						
R1504	00	Byte-4																																																						
R1505	00	Byte-5																																																						
R1506	00	Byte-6																																																						
R1507	00	Byte-7																																																						
R1508	00	Byte-8																																																						