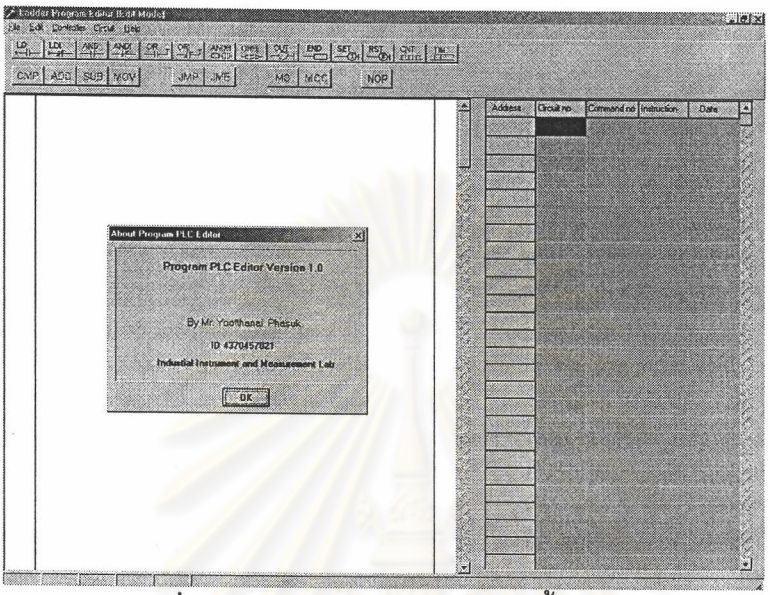
พลตรีจุลจักรกลาย สถาพันวิทยเดช
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6

ซอฟต์แวร์พัฒนาโปรแกรมขั้นบันได

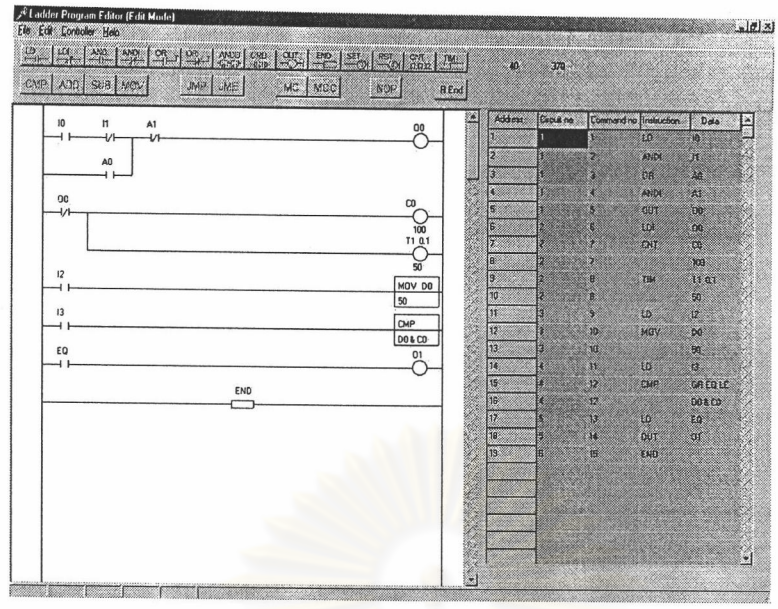


รูปที่ 6.1 ซอฟต์แวร์พัฒนาโปรแกรมขั้นบันได

ซอฟต์แวร์พัฒนาโปรแกรมขั้นบันไดทำหน้าที่ในการสร้าง/แก้ไข และเก็บบันทึกโปรแกรมขั้นบันไดที่ผู้ใช้เขียนขึ้น เป็นคอมไพเลอร์สำหรับแปลงโปรแกรมขั้นบันไดให้เป็นภาษาเครื่องที่ PLC เข้าใจ โหลดภาษาเครื่องของโปรแกรมขั้นบันไดลงไปในหน่วยความจำโปรแกรมภายนอกของ PLC และซอฟต์แวร์นี้ยังสามารถติดต่อกับ PLC เพื่อรับสถานะของรีเลย์ภายใน PLC มาแสดงผล ซอฟต์แวร์ถูกออกแบบให้ใช้งานง่าย ดำเนินการอยู่บนระบบปฏิบัติการ Windows พัฒนาขึ้นโดยใช้ภาษา Delphi

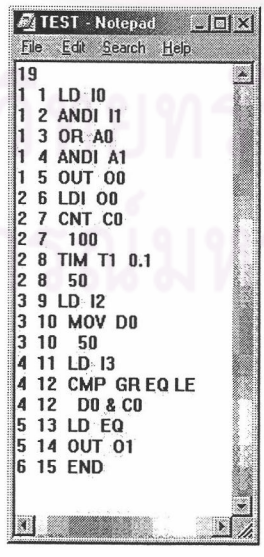
6.1 การสร้าง/แก้ไขและเก็บบันทึกโปรแกรมขั้นบันได

ซอฟต์แวร์ที่พัฒนาขึ้นนี้มีความสามารถในการสร้างโปรแกรมขั้นบันได มีทุกคำสั่งที่ PLC ในวิทยานิพนธ์นี้มี เมื่อทำการสร้างโปรแกรมขั้นบันไดทางด้านซ้ายของหน้าจอจะแสดงวงจรของโปรแกรมขั้นบันได และหน้าจอทางด้านขวาจะแสดงโปรแกรมขั้นบันไดในลักษณะภาษานิวโมนิค (Mnemonic) พร้อมแสดงตำแหน่ง Circuit No. และ Command no. เพื่อช่วยในการแก้ไขโปรแกรมภายหลัง การป้อนโปรแกรมออกแบบให้ผู้ใช้สามารถคลิกเลือกคำสั่งที่อยู่ด้านบนที่ละขั้น หลังนั้นทั้งสองส่วน โดยด้านขวาที่ก็จะแสดงคำสั่งในรูปภาษานิวโมนิคและทางด้านซ้ายก็จะแสดงในรูปวงจรขั้นไดที่เพิ่มขึ้น ดังแสดงในรูปที่ 6.2



รูปที่ 6.2 การสร้าง แก้ไขโปรแกรมขั้นบันได

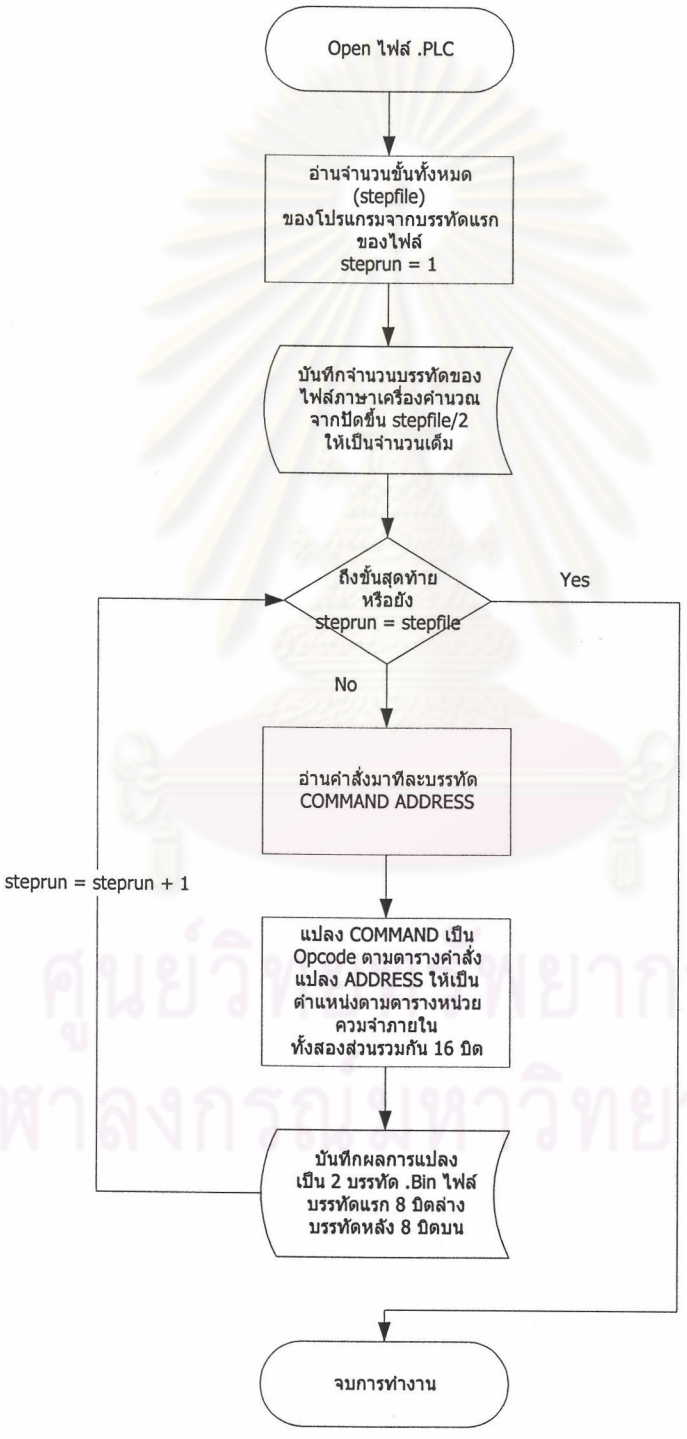
เมื่อสร้างโปรแกรมขั้นบันไดเสร็จเรียบร้อยแล้ว การจัดเก็บกำหนดให้บันทึกไฟล์ให้มีนามสกุลเป็น .PLC ซึ่งรูปแบบของไฟล์ถูกกำหนดขึ้นเองดังแสดงในรูปที่ 6.3 การบันทึกโปรแกรมขั้นบันไดจะเก็บในลักษณะภาษานิวโมนิคที่ตรงกับส่วนที่แสดงอยู่ในหน้าจอด้านขวา โดยบรรทัดแรกจะแสดงจำนวนขั้น (Step) ของโปรแกรมทั้งหมด และบรรทัดต่อๆ ไปก็จะเป็นส่วนของโปรแกรม ตัวเลขชุดแรกของบรรทัดคือ Circuit No ตัวเลขชุดที่สองคือ Command No. จากนั้นจึงเป็นคำสั่ง ลักษณะคล้ายกันจนถึงคำสั่งสุดท้าย และโปรแกรมที่แสดงนี้กำหนดให้ชื่อ TEST.PLC



รูปที่ 6.3 รูปแบบไฟล์ที่จัดเก็บนามสกุล PLC

6.2 การแปลงโปรแกรมขั้นบันไดเป็นภาษาเครื่อง

เมื่อโปรแกรมขั้นบันไดได้ถูกเขียนและจัดเก็บเสร็จแล้ว ขั้นตอนต่อไปซอฟต์แวร์พัฒนาโปรแกรมขั้นบันไดต้องมีความสามารถในการแปลงโปรแกรมขั้นบันไดที่ถูกจัดเก็บในรูปแบบภาษานิวโมนิคดังรูปที่ 6.3 ให้เป็นภาษาเครื่องที่ PLC เข้าใจได้ ซึ่งได้จากการกำหนด Instructions Machine, Opcode ของแต่ละคำสั่ง และตำแหน่งของข้อมูลที่คำสั่งเหล่านั้นเกี่ยวข้องด้วย การทำงานแสดงไว้ดังโฟล์วชาร์ตในรูปที่ 6.4



รูปที่ 6.4 โฟล์วชาร์ตการเปลี่ยนภาษานิวโมนิคจากไฟล์ .PLC เป็นภาษาเครื่องไฟล์ .BIN

Normal Instructions Machine

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Op-code				Address								bitno			

Expand 1 Instructions Machine

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	Op-code expand				Address							

Expand 2 Instructions Machine

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	Op-code expand							

ก. รูปแบบ Instructions Machine ทั้ง 3 แบบ

Normal PLC command

NO	Opcode	Mnemonic	Symbol
0	0000	LD	
1	0001	LDI	
2	0010	AND	
3	0011	ANDI	
4	0100	OR	
5	0101	ORI	
6	0110	OUT	
7	0111	SET	1
8	1000	RST	0
9	1001	TIM	TIM
A	1010	CNT	CNT
B	1011		
C	1100		
D	1101		
E	1110		
F	1111	EXPAND	

Expand 1 PLC command

NO	Opcode	Mnemonic	Symbol
0	0000	CMP	CMP
1	0001	ADD	ADD
2	0010	SUB	SUB
3	0011	MOV	MOV
4	0100		
5	0101		
6	0110		
7	0111		
8	1000		
9	1001		
A	1010		
B	1011		
C	1100		
D	1101		
E	1110		
F	1111	EXPAND	

Expand 2 PLC command

NO	Opcode	Mnemonic	Symbol
0	0000	ANDB	
1	0001	ORB	
2	0010	END	END
3	0011	MC	MC
4	0100	MCC	MCC
5	0101	JMP	JMP
6	0110	JME	JME
7	0111	NOP	NOP
8	1000		
9	1001		
A	1010		
B	1011		
C	1100		
D	1101		
E	1110		
F	1111		

ข. คำสั่งทั้งหมดแสดงค่า 0PCODE และรูปแบบ Instruction Machine

Data Memory address		Relay Number															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	INPUT	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
1		I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
2		I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
3		I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
4	OUTPUT	O15	O14	O13	O12	O11	O10	O9	O8	O7	O6	O5	O4	O3	O2	O1	O0
5		O31	O30	O29	O28	O27	O26	O25	O24	O23	O22	O21	O20	O19	O18	O17	O16
6	AUX	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
7		A31	A30	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16
8		A47	A46	A45	A44	A43	A42	A41	A40	A39	A38	A37	A36	A35	A34	A33	A32
9		A63	A62	A61	A60	A59	A58	A57	A56	A55	A54	A53	A52	A51	A50	A49	A48
10		A79	A78	A77	A76	A75	A74	A73	A72	A71	A70	A69	A68	A67	A66	A65	A64
11		A95	A94	A93	A92	A91	A90	A89	A88	A87	A86	A85	A84	A83	A82	A81	A80
12		A111	A110	A109	A108	A107	A106	A105	A104	A103	A102	A101	A100	A99	A98	A97	A96
13		A127	A126	A125	A124	A123	A122	A121	A120	A119	A118	A117	A116	A115	A114	A113	A112
14		A143	A142	A141	A140	A139	A138	A137	A136	A135	A134	A133	A132	A131	A130	A129	A128
15		A159	A158	A157	A156	A155	A154	A153	A152	A151	A150	A149	A148	A147	A146	A145	A144
16	SPARE														GR	EQ	LE
17																	

ค. ตำแหน่ง Input, Output, Aux และ Spare ในหน่วยความจำภายในส่วนข้อมูล

Data Memory address		Bit Number		
		15	14	13 - 0
18	Timer / Counter	O	LS	T/C 0
19		O	LS	T/C 1
20		O	LS	T/C 2
21		O	LS	T/C 3
22		O	LS	T/C 4
23		O	LS	T/C 5
24		O	LS	T/C 6
25		O	LS	T/C 7
26		O	LS	T/C 8
27		O	LS	T/C 9
28		O	LS	T/C 10
29		O	LS	T/C 11
30		O	LS	T/C 12
31		O	LS	T/C 13
32		O	LS	T/C 14
33		O	LS	T/C 15
34		O	LS	T/C 16
35		O	LS	T/C 17
36		O	LS	T/C 18
37		O	LS	T/C 19
38		O	LS	T/C 20
39		O	LS	T/C 21
40		O	LS	T/C 22
41		O	LS	T/C 23
42		O	LS	T/C 24
43		O	LS	T/C 25
44		O	LS	T/C 26
45		O	LS	T/C 27
46		O	LS	T/C 28
47		O	LS	T/C 29
48		O	LS	T/C 30
49		O	LS	T/C 31
50		O	LS	T/C 32
51		O	LS	T/C 33
52		O	LS	T/C 34
53		O	LS	T/C 35
54		O	LS	T/C 36
55		O	LS	T/C 37
56		O	LS	T/C 38
57		O	LS	T/C 39
58		O	LS	T/C 40
59		O	LS	T/C 41
60		O	LS	T/C 42
61		O	LS	T/C 43
62		O	LS	T/C 44
63		O	LS	T/C 45
64		O	LS	T/C 46
65		O	LS	T/C 47
66		O	LS	T/C 48
67		O	LS	T/C 49
68		O	LS	T/C 50
69		O	LS	T/C 51
70		O	LS	T/C 52
71		O	LS	T/C 53
72		O	LS	T/C 54
73		O	LS	T/C 55
74		O	LS	T/C 56
75		O	LS	T/C 57
76		O	LS	T/C 58
77		O	LS	T/C 59

Data Memory address		Bit Number		
		15	14	13 - 0
78	DATA	N/A	N/A	D0
79		N/A	N/A	D1
80		N/A	N/A	D2
81		N/A	N/A	D3
82		N/A	N/A	D4
83		N/A	N/A	D5
84		N/A	N/A	D6
85		N/A	N/A	D7
86		N/A	N/A	D8
87		N/A	N/A	D9
88		N/A	N/A	D10
89		N/A	N/A	D11
90		N/A	N/A	D12
91		N/A	N/A	D13
92		N/A	N/A	D14
93		N/A	N/A	D15
94		N/A	N/A	D16
95		N/A	N/A	D17
96		N/A	N/A	D18
97		N/A	N/A	D19
98		N/A	N/A	D20
99		N/A	N/A	D21
100		N/A	N/A	D22
101		N/A	N/A	D23
102		N/A	N/A	D24
103		N/A	N/A	D25
104		N/A	N/A	D26
105		N/A	N/A	D27
106		N/A	N/A	D28
107		N/A	N/A	D29
108		N/A	N/A	D30
109		N/A	N/A	D31
110		N/A	N/A	D32
111		N/A	N/A	D33
112		N/A	N/A	D34
113		N/A	N/A	D35
114		N/A	N/A	D36
115		N/A	N/A	D37
116		N/A	N/A	D38
117		N/A	N/A	D39
118		N/A	N/A	D40
119		N/A	N/A	D41
120		N/A	N/A	D42
121		N/A	N/A	D43
122		N/A	N/A	D44
123		N/A	N/A	D45
124		N/A	N/A	D46
125		N/A	N/A	D47
126		N/A	N/A	D48
127		N/A	N/A	D49

O : Output Result

LS : Last Status

N/A : Not Available

ง. ตำแหน่ง Timer/Counter และ Data register ในหน่วยความจำภายในส่วนข้อมูล

รูปที่ 6.5 ข้อมูลทั้งหมดที่ใช้ในการเปลี่ยนภาษานิวโมนิคเป็นภาษาเครื่อง

อธิบายการทำงานโฟล์วชาร์ทรูปที่ 6.4 ได้ดังนี้

1. ทำการเปิดไฟล์นามสกุล .PLC ที่ต้องการเปลี่ยนให้เป็นไฟล์ภาษาเครื่องนามสกุล .BIN
2. อ่านข้อมูลบรรทัดแรกของไฟล์นามสกุล .PLC ซึ่งจะเป็นจำนวนบรรทัดหรือขั้นของโปรแกรมขั้นบันไดนั้น กำหนดให้ steprun ซึ่งเป็นตัวชี้บรรทัดให้เป็นหนึ่ง
3. คำนวณจำนวนบรรทัดของไฟล์ภาษาเครื่องโดยหารสองปัดให้เป็นจำนวนเต็ม เนื่องจากการส่งข้อมูลระหว่างคอมพิวเตอร์และ PLC ส่งกันทีละ 8 บิต แต่จำนวนขั้นสูงสุดของ PLC นี้มี 384 ขั้นหรือ 768 words การส่งจำนวนบรรทัดด้วยข้อมูลแค่ 8 บิตโดยตรงจึงไม่พอ จึงต้องทำการหารสองปัดให้เป็นจำนวนเต็มช่วย เมื่อ PLC รับค่าแล้วทำการคูณด้วย 4 ก็จะเป็นจำนวนข้อมูลที่จะมีการโหลดจริง คำนวณเสร็จแล้วบันทึกลงเป็นบรรทัดแรกของไฟล์นามสกุล .BIN
4. ตรวจสอบว่าอ่านไฟล์มาทำการแปลงจนถึงบรรทัดสุดท้ายหรือยัง steprun = stepfile
5. ถ้ายังไม่ถึงก็ทำการอ่านคำส่งมาทีละบรรทัด ซึ่งจะแยกได้เป็น 2 ส่วนคือ COMMAND และ ADDRESS
6. จากรูปที่ 6.5 (ข) จะสามารถรู้ได้ว่า COMMAND นี้มี Instructions Machine แบบใดและมี Opcode เท่ากับเท่าไร และจากรูปที่ 6.5 (ค) หรือ (ง) จะสามารถรู้ได้ว่า ADDRESS นี้คืออะไรมีตำแหน่งในหน่วยความจำภายในส่วนข้อมูลเป็นเท่าไร และมี BITNO เป็นเท่าไรด้วย
7. จากรูปที่ 6.5 (ก) เมื่อรู้ว่าคำสั่งมี Instructions Machine แบบไหน มี Opcode เท่าไร และ ADDRESS อยู่ตำแหน่งไหนแล้วก็ทำการแทนค่าตามรูปแบบรูปที่ 6.5 (ก) จะได้ผลการแปลงของภาษาเครื่องขนาด 16 บิต
8. จากรูปที่ 4.6 เนื่องหน่วยความจำภายนอกที่ใช้ EEPROM อ่าน/เขียนข้อมูลทีละ 8 บิต ดังนั้นผลจากการแปลงในข้อ 7. จะทำการแยกเขียนในไฟล์นามสกุล .BIN เป็น 2 บรรทัด โดยบรรทัดแรกคือผลการแปลง 8 บิตล่าง และบรรทัดหลังคือผลการแปลง 8 บิตบน
9. จากนั้นทำการเพิ่มตัวอ้างตำแหน่งบรรทัดขึ้นอีกหนึ่ง แล้วไปทำข้อ 4. ใหม่
10. เมื่อทำการแปลงถึงบรรทัดสุดท้าย steprun = stepfile ก็ถือว่าสิ้นสุดกระบวนการ

ตัวอย่างการแปลง

นำ 3 บรรทัดตัวอย่างจากโปรแกรม TEST.PLC มาแสดงการแปลงให้เป็นภาษาเครื่องได้ดังต่อไปนี้

1. การแปลงคำสั่งในกลุ่ม Normal Instruction Machine

จาก Command No. 1 1 1 LD I0

OPCODE ของ LD คือ 0000

ADDRESS ของ I0 คือ 0000 0000

Bitno ของ I0 คือ 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Op-code				Address								bitno			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

เก็บในไฟล์นามสกุล .BIN แยกได้เป็น 2 บรรทัดคือ

แยกเก็บ 8 บิตล่าง 0000 0000 = 0 ฐานสิบ

แยกเก็บ 8 บิตบน 0000 0000 = 0 ฐานสิบ

2. การแปลงคำสั่งในกลุ่ม Expand 1 Instruction Machine

จาก Command No. 12 4 12 CMP

5 12 D0 & CO

ชุดคำสั่งนี้มีความยาว 2 ขั้น (Step) หรือ 2 words

ขั้นที่หนึ่ง

OPCODE Expand ของ CMP คือ 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	Op-code expand				Address							
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

เก็บในไฟล์นามสกุล .BIN แยกได้เป็น 2 บรรทัดคือ

แยกเก็บ 8 บิตล่าง 0000 0000 = 0 ฐานสิบ

แยกเก็บ 8 บิตบน 1111 0000 = 240 ฐานสิบ

ขั้นที่สอง

ADDRESS ของ C0 คือ 0001 0010

ADDRESS ของ D0 คือ 0100 1110

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	0	0	1	0	0	1	0

แยกเก็บ 8 บิตล่าง 0001 0010 = 18 ฐานสิบ

แยกเก็บ 8 บิตบน 0100 1110 = 78 ฐานสิบ

3. การแปลงคำสั่งในกลุ่ม Expand 2 Instruction Machine

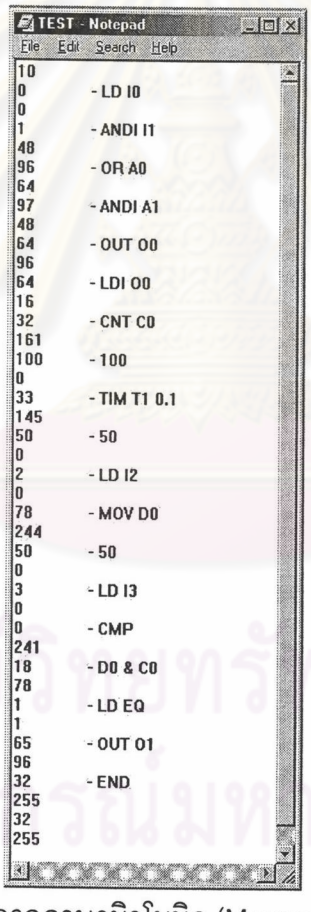
จาก Command No. 15 6 15 END

OPCODE Expand2 ของ END คือ 0010

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	Op-code expand							
1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	0

แยกเก็บ 8 บิตล่าง 0010 0000 = 32 ฐานสิบ

แยกเก็บ 8 บิตบน 1111 1111 = 255 ฐานสิบ

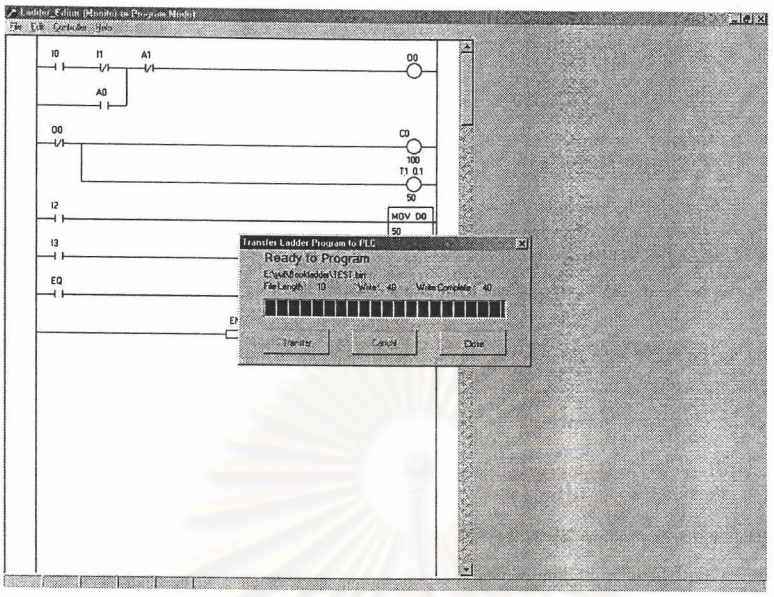


รูปที่ 6.6 ผลการเปลี่ยนจากภาษานิวโมนิค (Mnemonic) ให้เป็นภาษาเครื่อง

ซอฟต์แวร์พัฒนาโปรแกรมขั้นบันไดจะเปลี่ยนโปรแกรมขั้นบันไดในรูปภาษานิวโมนิค (Mnemonic) ที่อยู่ในไฟล์นามสกุล PLC ไปเป็นภาษาเครื่องที่ PLC เข้าใจ ได้ผลลัพธ์อยู่ในไฟล์นามสกุล BIN แสดงในรูปที่ 6.6 ที่พร้อมจะสามารถโหลดลง PLC ได้ต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

6.3 โหลดภาษาเครื่องของโปรแกรมขั้นบันไดลง PLC



รูปที่ 6.7 การโหลดโปรแกรมภาษาเครื่องลง PLC

การโหลดภาษาเครื่องที่อยู่ในไฟล์นามสกุล BIN จากคอมพิวเตอร์เพื่อเขียนลงในหน่วยความจำโปรแกรมภายนอกของ PLC ก่อนอื่นจะต้องส่งจำนวนข้อมูลโปรแกรมที่จะโหลดไปก่อน แต่เนื่องจากหน่วยความจำภายในส่วนโปรแกรม (Program Memory) มีขนาดสูงสุด 384 ขั้น (Step) ซึ่งเท่ากับ 384×16 บิต หรือเท่ากับ 768 Byte แต่การส่งข้อมูลสามารถทำได้ครั้งละ 8 บิต ดังนั้นจำนวนข้อมูลที่จะส่งจึงคำนวณได้จากจำนวนบรรทัดของคำสั่งที่แสดงในรูป 6.3 ซึ่งเท่ากับ 19 หารด้วย 2 ผลลัพธ์ที่ได้ปัดขึ้นให้เป็นจำนวนเต็มได้เป็น 10 แสดงในบรรทัดแรกของไฟล์นามสกุล BIN ในรูปที่ 6.6 จำนวนข้อมูลที่จะส่งจริงก็คือ 19×2 เท่ากับ 38 เพิ่มคำสั่ง END เข้าไปอีก 2 กลายเป็น 40 ดังนั้นเมื่อ PLC รับค่า 10 เข้าไปก็จะทำการคูณด้วย 4 เท่ากับ 40 ก็จะเป็นจำนวน Byte ของข้อมูลที่จะมีการโหลดจริง การทำเช่นนี้เพื่อเป็นการประหยัด State Machine ใน PLC

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

6.4 แสดงสถานะของรีเลย์ภายใน PLC

99	100	Data Address	Data Low Byte	Data High Byte	Zero Correction

รูปที่ 6.8 รูปแบบการส่งข้อมูลในหน่วยความจำภายในส่วนข้อมูลไปคอมพิวเตอร์ส่วนบุคคล

ข้อมูลสถานะรีเลย์ภายในที่อยู่ในหน่วยความจำภายในส่วนข้อมูลมีทั้งหมด 128 ตำแหน่งดังแสดงให้รูปที่ 4.2 โดยทุกรอบการทำงาน 1 Scan Time PLC จะตรวจสอบสถานะของ Transmitter Module ถ้าพร้อมก็จะส่งข้อมูลมายังคอมพิวเตอร์ทีละตำแหน่งวนรอบไปเรื่อยๆ รูปแบบการส่งข้อมูลแสดงในรูปที่ 6.8 ซอฟต์แวร์พัฒนาโปรแกรมขั้นบันไดจะอ่านข้อมูลนี้เป็นชุดดังรูปที่ 6.9 แล้วซอฟต์แวร์พัฒนาโปรแกรมขั้นบันไดจะทำการแปลงความหมายและแสดงผล

รูปแบบการส่งข้อมูล PLC จะส่ง 99 แล้วตามด้วย 100 จากนั้นก็เป็น Data Address ตามด้วย Data Low Byte และ Data High Byte สุดท้ายคือ Zero Correction ซึ่งมีเพิ่มเข้ามาเพื่อแก้ปัญหาที่ซอฟต์แวร์พัฒนาโปรแกรมขั้นบันไดอ่านข้อมูลจากพอร์ตอนุกรมโดย Function Get Char ไม่สามารถอ่านข้อมูลที่มีค่าเท่ากับศูนย์ที่ถูกส่งมาโดย PLC ได้ ดังนั้นเมื่อข้อมูลไม่ว่าจะเป็นส่วน Data address, Data Low byte หรือ Data High byte เท่ากับศูนย์ PLC จะส่งหนึ่งมาแทน แล้วบอกให้ส่วนที่รับข้อมูลที่อยู่ซอฟแวร์ในคอมพิวเตอร์รู้โดยส่งข้อมูล Zero Correction เพิ่มเข้ามา บิตที่ 7 กำหนดให้เป็นหนึ่งเสมอ ถ้า Address เป็น 0 การส่งจะส่ง 1 ออกมาแทนและกำหนดบิตที่ 6 ของ Zero Correction ให้เป็น 1 ถ้า Data Low Byte เป็น 0 จะส่งหนึ่งออกมาแทนและกำหนดบิตที่ 5 ของ Zero Correction ให้เป็น 0 และถ้า Data High Byte เป็น 0 จะส่งหนึ่งออกมาแทนและกำหนดให้บิตที่ 4 ของ Zero Correction จะถูกกำหนดให้เป็น 1 ดังแสดงในตัวอย่าง

ตัวอย่าง

ถ้า PLC ต้องการส่ง Address 0 Data Low Byte 0 Data High Byte 0 รูปแบบการส่งก็จะเป็น

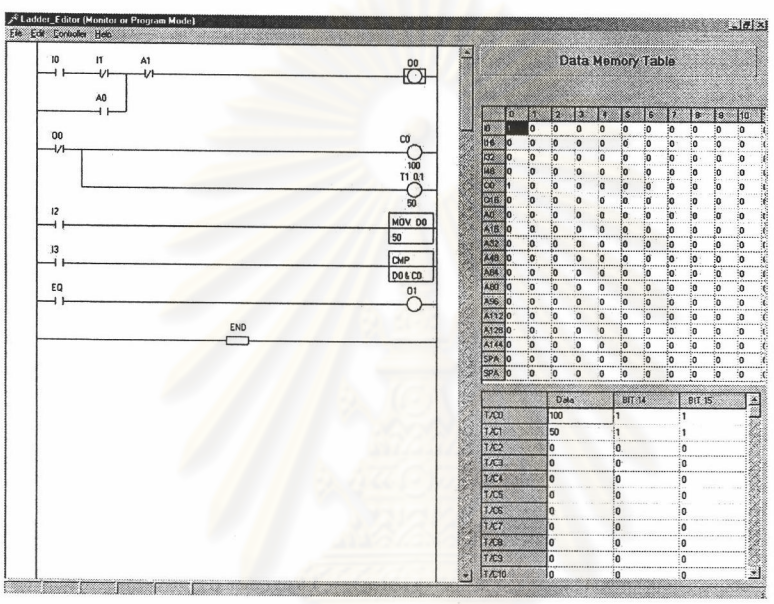
99	100	1	1	1	240

โดย Zero Correction จะเท่ากับ 1111 0000 เท่ากับ 240 ฐานสิบ

ถ้า PLC ต้องการส่ง Address 56 Data Low Byte 100 Data High Byte 0

99	100	56	100	1	144

โดย Zero Correction จะเท่ากับ 1001 0000 เท่ากับ 144 ฐานสิบ



รูปที่ 6.9 หน้าต่างแสดงผลหน่วยความจำภายในของ PLC

ขณะที่เครื่อง PLC ทำกำลังทำงานควบคุมอยู่ในโหมดทำงานส่งสถานะรีเลย์ภายในมายังคอมพิวเตอร์ โดยซอฟแวร์พัฒนาโปรแกรมขั้นบันไดนี้สามารถอ่านค่าทำการแปลงความหมายแล้วแสดงผลให้ผู้ใช้ทราบ ดังแสดงในรูปที่ 6.9

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย