

Creating a PLC Control Program with Conditional Step Diagram Method

Chalermchon Visayadamrong

<http://eprints.utcc.ac.th/id/eprint/927>

การสร้างโปรแกรมควบคุม PLC ด้วยวิธี Conditional Step Diagram

เฉลิมชนม์ ไชยดำรง

ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์

126/1 ถนนวิภาวดีรังสิต เขตดินแดง กรุงเทพฯ 10600

โทรศัพท์: 0-26976705 โทรสาร: 0-2275-4892, Email: chalemchon_vis@utcc.ac.th

คำสำคัญ : การจัดวางบรรจภูณฑ์

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อสร้างรูปแบบการวิเคราะห์ และขั้นตอนการออกแบบโปรแกรมควบคุม PLC ในลักษณะ Conditional Step Diagram เนื่องจากประสบการณ์การเป็นผู้สอนในวิชา Automation พบว่าการเขียนโปรแกรมควบคุม PLC นั้นมีลักษณะในการวิเคราะห์เงื่อนไขแตกต่างจากการเขียนโปรแกรมทั่วไป อาทิเช่น ภาษาซี เป็นต้น เนื่องจากขั้นตอนการทำงานของโปรแกรมควบคุม PLC นั้นมีการตรวจสอบสถานการณ์ทำงานของอุปกรณ์อินพุตที่เชื่อมต่อทั้งหมด แล้วจึงนำผลที่ตรวจสอบได้มาประมวลผลหาผลลัพธ์จากโปรแกรมที่ใช้ในการควบคุมผลลัพธ์ที่ได้จากการประมวลผลจะถูกนำไปสั่งให้อุปกรณ์เอาต์พุตทำงานตามเงื่อนไขของโปรแกรม ซึ่งขั้นตอนทั้งหมดใช้เวลาในการทำงานเพียง 1 ใน 1000 วินาที ซึ่งแตกต่างจากการเขียนโปรแกรมทั่วไป ตัวอย่างเช่น เมื่อมีคำสั่งในการรับค่าจากผู้ใช้งาน ลักษณะของโปรแกรมทั่วไปจะหยุดรอจนกว่าผู้ใช้งานจะป้อนข้อมูล แล้วจึงทำงานในคำสั่งถัดไป แต่การเขียนโปรแกรมควบคุม PLC ในลักษณะเดียวกันนี้จะแตกต่างโดยสิ้นเชิง คือ ถ้าผู้ใช้งานไม่มีการป้อนข้อมูลใด จะมีการทำงานในคำสั่งถัดไปทันที ดังนั้นในการเขียนโปรแกรมควบคุม PLC ในลักษณะเงื่อนไขนั้นจะมีการสร้างตัวแปรภายในเพื่อควบคุมลำดับการทำงานในแต่ละขั้นตอน ซึ่งผู้ที่มีความชำนาญในการเขียนโปรแกรมควบคุม PLC จะทราบลักษณะการทำงานเช่นนี้เป็นอย่างดี และมีแนวทางเขียนโปรแกรมเพื่อใช้ในการควบคุมตามที่ตนเองออกแบบแตกต่างกัน โดยไม่มีรูปแบบและวิธีการที่ชัดเจน ซึ่งจะแตกต่างกันตามทักษะของผู้เขียนโปรแกรม ซึ่งจากปัญหานี้ทำให้นักศึกษาที่เริ่มเรียนรู้การเขียนโปรแกรม PLC ใหม่จะสับสน ดังนั้นจากประสบการณ์การสอนได้รวบรวมปัญหาที่พบจากการสอนและการออกแบบในงานโครงการต่าง ๆ แล้วนำมาสร้างรูปแบบการวิเคราะห์และขั้นตอนในการออกแบบโปรแกรมควบคุม PLC ในลักษณะ Conditional Step Diagram ที่เป็นระบบ จากผลการทดสอบรูปแบบดังกล่าว พบว่าผลลัพธ์ที่ได้จากวิธีการนี้สามารถนำมาสร้างโปรแกรมควบคุม PLC ได้อย่างถูกต้อง และง่ายต่อการสอนนักศึกษา รวมถึงสามารถใช้ในการออกแบบโปรแกรมควบคุม PLC ที่ซับซ้อนได้เป็นอย่างดี

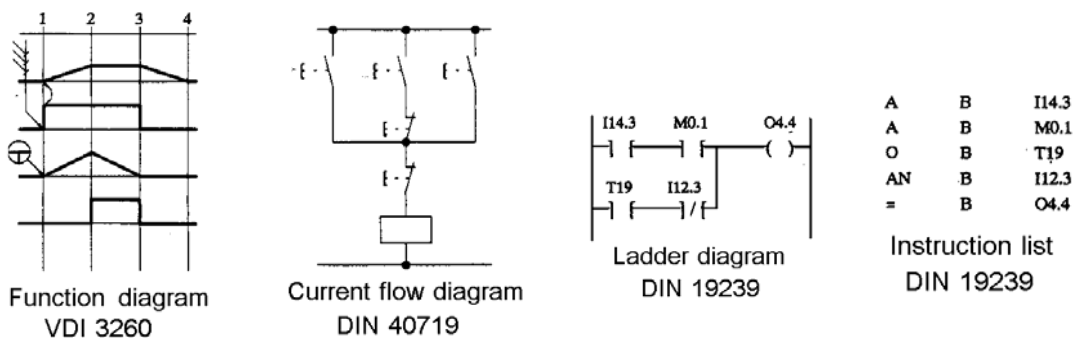
1. บทนำ

แม้ว่าในปัจจุบันการควบคุมกระบวนการผลิตนั้นมีการนำระบบคอมพิวเตอร์ที่มีความสามารถในการควบคุมและเก็บข้อมูลของการปฏิบัติงานของเครื่องจักร หรือที่เรียกว่า ระบบสกาตา (SCADA , Supervisory Control And Data Acquisition) แต่พื้นฐานที่สำคัญที่ใช้ในการควบคุมเครื่องจักรในโรงงานนั้นยังคงใช้ PLC (Programmable

Logic Controller) เนื่องจากมีเสถียร และทนทานต่อสภาวะที่เลวร้ายได้ดีกว่าเครื่องคอมพิวเตอร์ โดยหัวใจสำคัญในการควบคุมเครื่องจักรก็ยังคงอาศัย PLC เป็นหลัก โดย PLC จะมีความสามารถในการติดต่อสื่อสารกับระบบคอมพิวเตอร์ที่ใช้ในการควบคุมจากระยะไกล รวมทั้งในปัจจุบัน PLC มีความสามารถสูงขึ้นและราคาถูกลง ทำให้มีการใช้งานในภาคอุตสาหกรรมยิ่งแพร่หลาย ซึ่งในการเรียนการสอนในหลักสูตรวิศวกรรมอุตสาหกรรมและวิศวกรรมเครื่องกล จึงมีวิชาที่เกี่ยวข้องกับระบบควบคุมอัตโนมัติเป็นวิชาแกน จากประสบการณ์การสอนวิชาระบบควบคุมอัตโนมัติ พบว่าในการออกแบบระบบควบคุมด้วยอุปกรณ์ระบบนิวแมติกส์และไฮดรอลิกส์ด้วย PLC นั้น ยังไม่มีวิธีการออกแบบที่เป็นรูปแบบชัดเจน ในการเรียนการสอนส่วนใหญ่จะเน้นที่รูปแบบคำสั่งและการควบคุมที่ไม่ซับซ้อน จากปัญหาในการสอนเขียนโปรแกรมควบคุมด้วย PLC พบว่านักศึกษาส่วนใหญ่เกิดความสับสน เนื่องจากลักษณะของการทำงานของ PLC นั้นมีความแตกต่างจากหลักการเขียนโปรแกรมทั่วไปอย่างเช่น ภาษา VB และ ภาษา C เป็นต้น ดังนั้นงานวิจัยนี้จึงได้พัฒนาขั้นตอนในการออกแบบวิเคราะห์สัญญาณจากอุปกรณ์ตรวจจับตำแหน่งของอุปกรณ์ทำงาน และการสร้างสัญญาณควบคุม ที่สามารถนำไปใช้ในการออกแบบระบบควบคุมด้วยระบบนิวแมติกส์ และระบบไฟฟ้าได้ รวมทั้งการควบคุมด้วย PLC เพื่อง่ายต่อการศึกษาและสามารถนำมาใช้ในการออกแบบที่ซับซ้อนได้ ซึ่งจะกล่าวในหัวข้อถัดไป

2. ทฤษฎีที่เกี่ยวข้อง

ในการออกแบบการทำงานอุปกรณ์ที่ควบคุมด้วย PLC นั้นจะมีเครื่องมือในการอธิบายการทำงานและรูปแบบการควบคุมหลายรูปแบบ แต่ที่นิยมและใช้งานอย่างแพร่หลาย ได้แก่ การอธิบายรูปแบบการทำงานของอุปกรณ์กำลังด้วย แผนภาพการเคลื่อนที่ (Function diagram) การเขียนวงจรไฟฟ้าควบคุมด้วยแผนภาพหน้าสัมผัส (Current flow diagram) การเขียนโปรแกรมควบคุม PLC ในรูปแบบของเลดเดอร์ไดอะแกรม และในรูปแบบโครงสร้างภาษาอินสตรัคชันลิส (Instruction list) ซึ่งสามารถใช้รูปแบบใดรูปแบบหนึ่งได้ หรือใช้ร่วมกัน ซึ่งทั้งหมดที่กล่าวมาเป็นรูปแบบตามมาตรฐานสากลดังแสดงตัวอย่างและรหัสของมาตรฐานในรูปที่ P2



รูปที่ P2 แสดงตัวอย่างการใช้ไดอะแกรมในการออกแบบการควบคุมด้วย PLC และรหัสมาตรฐาน

2.1 โครงสร้างและลำดับการทำงานภายใน PLC

องค์ประกอบของ PLC โดยทั่วไปแสดงดังรูปที่ P3 โดยประกอบด้วย 2 ส่วนหลักดังนี้

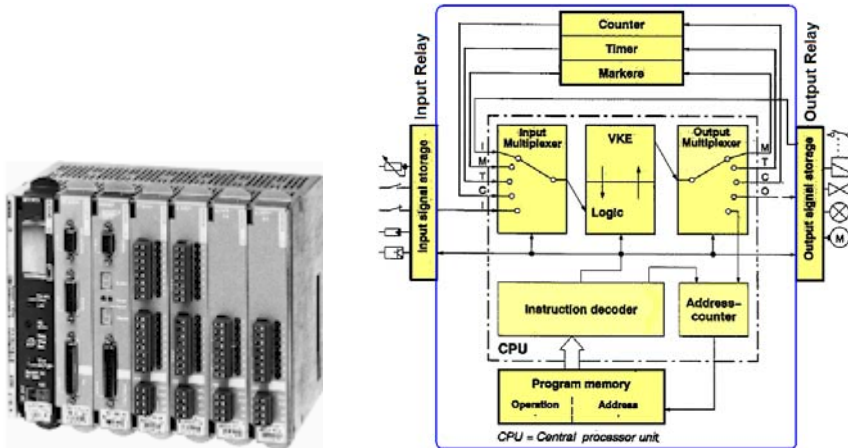
1) ส่วนที่ใช้ในการติดต่ออุปกรณ์ภายนอก

- ชุดหน้าสัมผัสของอินพุทรีเลย์ที่ใช้ในการตรวจจับสัญญาณที่เกิดจากอุปกรณ์ตรวจจับ (Sensor) ตัวอย่างอุปกรณ์ตรวจจับ เช่น สวิตช์รูปแบบต่าง ๆ ทั้งชนิดสัมผัสและไม่สัมผัส

- ชุดเอาพุตรีเลย์ที่ใช้ในการจ่ายกระแสไฟให้กับอุปกรณ์ทำงานต่าง ๆ ภายนอก ตัวอย่างอุปกรณ์ทำงาน เช่น หลอดไฟ มอเตอร์ วาล์วควบคุมการไหล หน้าสัมผัสรีเลย์ เป็นต้น

2) ส่วนประมวลผลภายใน

ได้แก่ หน่วยประมวลผล(CPU)และหน่วยความจำ(Address)สำหรับการประมวลโปรแกรม โดยผลลัพธ์ที่ได้จะสั่งงานให้หน้าสัมผัสเกิดการเปลี่ยนแปลงส่งผลให้อุปกรณ์ควบคุมที่ต่อเชื่อมกับ PLC เกิดการทำงานตามโปรแกรมที่กำหนด โดยหน่วยความจำประกอบด้วย หน่วยความจำหน้าสัมผัสอินพุท หน้าสัมผัสเอาพุท และหน้าสัมผัสเสมือน (Markers) ซึ่งมีค่าสถานะ เป็น ON หรือ OFF หน่วยความจำตัวนับ(Counter) และตัวตั้งเวลา(Timer) เป็นชนิดตัวเลข



รูปที่ P3 แสดงตัวอย่าง PLC รุ่น CL200 ยี่ห้อ Bosch และรูปโครงสร้างการทำงานภายใน

2.2 กระบวนการทำงานภายใน PLC

การทำงาน 1 รอบโปรแกรมของ PLC มีลำดับทำงานของหน่วยประมวลผลดังนี้

1. อ่านค่าสถานะอินพุทรีเลย์และเอาพุทรีเลย์ (Scan input/output Multiplexer) เพื่อนำไปกำหนดค่าให้กับหน่วยความจำภายใน PLC เพื่อใช้ในการประมวลผลในขั้นต่อไป
2. การประมวลผลโปรแกรม โดยนำค่าหน่วยความจำที่เป็นตัวแทนของสถานะอินพุท เอาพุท ตัวนับ ตัวตั้งเวลา และหน่วยความจำชั่วคราวภายใน PLC มาประมวลผลตามโปรแกรมที่ผู้ใช้กำหนด จัดเก็บผลลัพธ์ของโปรแกรมไว้ในหน่วยความจำเอาพุท
3. เปลี่ยนสถานะเอาพุทรีเลย์ เพื่อจ่ายกระแสไฟให้กับอุปกรณ์ทำงานตามผลลัพธ์ที่ประมวลผลจากข้อ 2
4. วนกลับไปทำงานในข้อที่ 1 อีกครั้งเป็นวัฏจักรตลอดเวลาที่ PLC ยังคงทำงานอยู่ดังแสดงในรูปที่ P4

2.3 Scan Cycle Time (SCT)

หมายถึงระยะเวลาที่ PLC ทำงาน 1 รอบโปรแกรมห่วงข้อที่กล่าวมาแล้ว ซึ่งเป็นค่าที่ใช้ในการวัดประสิทธิภาพในการทำงานของ PLC ซึ่งโดยทั่วไปแล้วมีค่าประมาณ 10 ms ซึ่งเพียงพอต่อการใช้งานควบคุมระบบอุตสาหกรรมทั่วไป

แต่ในกรณีกระบวนการผลิตอุปกรณ์อิเล็กทรอนิกส์ที่ใช้หุ่นยนต์อุตสาหกรรมรวมด้วยนั้น อาจมีความเร็วในการทำงานหนึ่งน้อยกว่าค่า 10 ms อาจทำให้ PLC ทั่วไปอาจไม่สามารถทราบถึงการเปลี่ยนแปลงและไม่ตอบสนอง

เหตุการณ์ ในกรณีนี้จำเป็นต้องใช้ PLC ที่สามารถสูงและมีค่า SCT น้อยกว่าระยะที่น้อยสุดที่หุ่นยนต์นั้นทำงานได้จึงสามารถควบคุมกระบวนการได้อย่างเที่ยงตรง

2.4 การประมวลผลคำสั่ง

จากรูปที่ P4 แสดงลักษณะการทำงานของโปรแกรมที่อยู่ภายในหน่วยความจำของ PLC จะเริ่มต้นที่บรรทัดที่ 1 และทำการประมวลผลแต่ละคำสั่งเรียงต่อกันจนจบโปรแกรมที่คำสั่ง EP (End of Program) แล้ววนกลับไปเริ่มต้นที่บรรทัดที่ 1 อีกครั้งต่อเนื่องเป็นวัฏจักร ตัวอย่างของคำสั่งและลำดับการประมวลผลในแต่ละคำสั่งดังแสดงในรูปที่ p4 โดยเริ่มที่บรรทัดที่คำสั่ง “ A I0.01 “ (PA 47) เป็นการตรวจสอบสถานะของหน้าสัมผัสอินพุต I0.0 มีค่าเป็นจริงหรือหนึ่ง (ถ้าหน้าสัมผัสปิด) และมีค่าเป็นเท็จหรือศูนย์ (ถ้าหน้าสัมผัสเปิด) นำค่าที่ได้พักไว้ในหน่วยความจำ VKE แล้วประมวลผลคำสั่งถัดไป “ A I0.4 “ (PA 48) นำค่าสถานะของหน้าสัมผัสอินพุต I0.4 ไปเปรียบเทียบกับเงื่อนไขและกับค่าในหน่วยความจำ VKE แล้วนำผลลัพธ์ที่ได้ใหม่จัดเก็บไว้ในหน่วยความจำ VKE อีกครั้ง ทำเช่นนี้ต่อเนื่องไปจนถึงบรรทัดคำสั่ง “ = O2.3 “ (PA 51) เป็นคำสั่งในการกำหนดค่าให้เอาพุต O2.3 ด้วยค่าของผลลัพธ์ที่จัดเก็บในหน่วยความจำ VKE

	<table border="1"> <thead> <tr> <th>PA</th> <th>Instruction</th> <th>Input Status</th> <th>Logic operation</th> <th>VKE</th> <th>Output Status</th> </tr> </thead> <tbody> <tr> <td>47</td> <td>A I0.1</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>48</td> <td>A I0.4</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>49</td> <td>AN M4.3</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>50</td> <td>O O2.2</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>51</td> <td>= O2.3</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>52</td> <td>S M0.5</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>53</td> <td>AN M4.0</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	PA	Instruction	Input Status	Logic operation	VKE	Output Status	47	A I0.1					48	A I0.4					49	AN M4.3					50	O O2.2					51	= O2.3					52	S M0.5					53	AN M4.0						<table border="1"> <tr> <td>A</td> <td>-SW1</td> </tr> <tr> <td>O</td> <td>-Mb</td> </tr> <tr> <td>AN</td> <td>-SW2</td> </tr> <tr> <td>=</td> <td>-Mb</td> </tr> <tr> <td>A</td> <td>-Mb</td> </tr> <tr> <td>S</td> <td>-L1</td> </tr> <tr> <td>R</td> <td>-L1</td> </tr> <tr> <td>EP</td> <td>-L2</td> </tr> </table>	A	-SW1	O	-Mb	AN	-SW2	=	-Mb	A	-Mb	S	-L1	R	-L1	EP	-L2
PA	Instruction	Input Status	Logic operation	VKE	Output Status																																																														
47	A I0.1																																																																		
48	A I0.4																																																																		
49	AN M4.3																																																																		
50	O O2.2																																																																		
51	= O2.3																																																																		
52	S M0.5																																																																		
53	AN M4.0																																																																		
A	-SW1																																																																		
O	-Mb																																																																		
AN	-SW2																																																																		
=	-Mb																																																																		
A	-Mb																																																																		
S	-L1																																																																		
R	-L1																																																																		
EP	-L2																																																																		
<p>Scan Cycle Time</p>		<p>ก) เลดเดอร์ไดอะแกรม</p>	<p>ข) ภาษาอินสตรักชันลิส</p>																																																																

รูปที่ P4 แสดงลักษณะการประมวลผลในแต่ละคำสั่ง

7.5 โครงสร้างภาษาอินสตรักชัน Instruction List

จากการเขียนเลดเดอร์ไดอะแกรมสามารถนำมาสร้างเป็นภาษาอินสตรักชันลิส (Instruction List) ด้วยการแทนสัญลักษณ์หน้าสัมผัสด้วยคำสั่ง ตัวแสดงอยู่ในรูปที่ p5 ซึ่งประกอบด้วย

ชุดคำสั่งในการเปรียบเทียบค่าของหน้าสัมผัสของอินพุต (Compare Input) ได้แก่ คำสั่ง A(And) AN(And Not) O(Or) และ ON(Or Not) โดยถ้าหน้าสัมผัสต่ออนุกรมกันจะหมายถึงการนำค่าหน้าสัมผัสมา And กัน ถ้าต่อขนานหมายถึงนำค่าหน้าสัมผัสมา OR กัน ถ้าหน้าสัมผัสเป็นแบบปกติปิดจะหมายถึง Not

ชุดคำสั่งในการควบคุมหน้าสัมผัสเอาพุต (change Output) ได้แก่ คำสั่ง

= (Assign result) เป็นการกำหนดให้หน้าสัมผัสเอาพุตจ่ายกระแสไฟ ตามผลลัพธ์ของการเปรียบเทียบ

S(Set) เป็นการกำหนดให้หน้าสัมผัสเอาพุทจ่ายกระแสไฟถาวร โดยคำสั่ง = จะไม่มีผลต่อหน้าสัมผัสเอาพุทที่ถูกคำสั่งนี้ จนกว่าจะถูกยกเลิกด้วยคำสั่ง R

R(Reset) เป็นการกำหนดให้หน้าสัมผัสเอาพุทจ่ายยกเลิกการจ่ายกระแสไฟถาวร

เนื่องจากในการทำงานของ PLC นั้นจะมีการทำงานร่วมกับอุปกรณ์ภายนอก ดังนั้นในการทำงานจะใช้เวลาในการอ่านสถานะของอินพุทที่ต่อเชื่อม แล้วจึงนำมาประมวลผลกับโปรแกรมควบคุม เมื่อได้ผลลัพธ์จะนำไปเปลี่ยนสถานะของเอาพุท เพื่อจ่ายกระแสไฟไปเลี้ยงระบบการทำงาน ซึ่งหนึ่งรอบการทำงานนั้นจะต้องมีระยะเวลาสั้นกว่าระยะเวลาที่อุปกรณ์อินพุททำงาน มิฉะนั้น PLC จะตรวจไม่พบการเปลี่ยนแปลงของอุปกรณ์อินพุท

การออกแบบ

เป็นการออกแบบการทำงานให้ขั้นตอนย่อย ๆ ที่ทำงานต่อเนื่องกัน โดยมีกฎว่า ลำดับงานที่ต้องต่อเนื่องกันตามลำดับเท่านั้น ไม่สามารถสลับได้ การทำงานในแต่ละขั้นตอนย่อยจะสิ้นสุดก่อน ถึงจะทำงานในขั้นถัดต่อไป

Initial or Reset State

เป็นขั้นตอนในเตรียมพร้อมการทำงาน โดยมีขั้นตอนดังนี้ คือ สั่งให้อุปกรณ์ต่าง ๆ เคลื่อนที่อยู่ในตำแหน่งเริ่มต้นที่พร้อมเริ่มทำงาน รวมทั้งกำหนดค่าต่าง ๆ ในหน่วยความจำของ PLC ให้มีค่าเริ่มต้น หรือในกรณีเกิดปัญหาและระบบหยุดทำงาน จำเป็นต้องมีการเริ่มต้นการทำงานใหม่ ก็จะมีขั้นตอนเช่นเดียวกัน

Step Working

ในการทำงานหนึ่งขั้นตอนย่อยจะประกอบด้วย 3 ขั้นตอนตามลำดับดังนี้

- 1) การสั่งงาน Action State เป็นขั้นตอนการทำงานที่กำหนด เมื่อมีสัญญาณเริ่มต้นทำงาน
- 2) รอการทำงาน Waiting State เป็นช่วงเวลารอให้งานที่ถูกสั่งในขั้นตอนสั่งงาน ทำงาน
- 3) จบการทำงาน End State เป็นขั้นตอนในการทำงานที่กำหนด เมื่อการทำงานเสร็จสิ้นแล้ว เช่น เพื่อสร้างสัญญาณสั่งงานในลำดับถัดไป หรือ การเคลียร์ค่าต่าง ๆ เพื่อทำงานในลำดับถัดไป

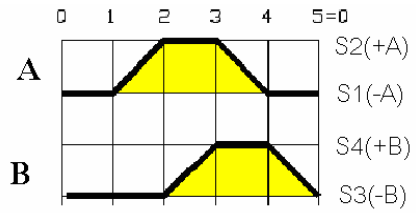
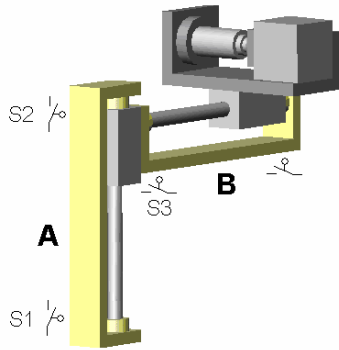
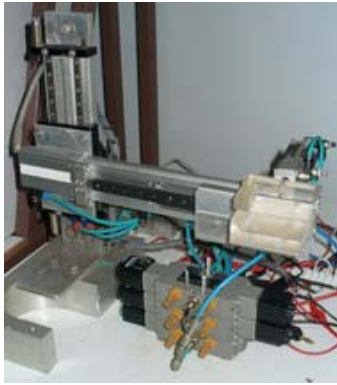
ตารางที่ 7.1 ตัวอย่างแสดงเหตุการณ์ในแต่ละงานย่อยจากรูปที่ 7.26

Action State	Waiting State	End State
เมื่อปุ่ม S1 ถูกกด สั่งให้ระบบอกสูบเคลื่อนที่ออก	ก้านสูบกำลังเคลื่อนที่ออก	รอจนก้านสูบเคลื่อนที่ออกสุด (ไม่มีคำสั่งทำงาน)
เมื่อปุ่ม S2 ถูกกด สั่งให้ระบบอกสูบเคลื่อนที่กลับ	ก้านสูบกำลังเคลื่อนที่เข้า	รอจนก้านสูบเคลื่อนที่เข้าสุด (ไม่มีคำสั่งทำงาน)

การประยุกต์ใช้ Step Function ในการออกแบบวงจรควบคุมที่ซับซ้อน

ขั้นตอนการทำงาน

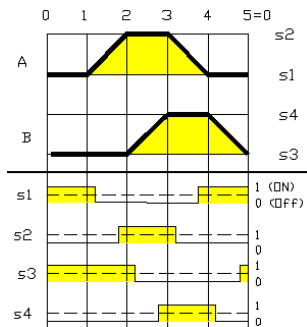
การออกแบบขั้นตอนการทำงาน โดยวาดภาพอธิบายรูปแบบการเคลื่อนที่ของอุปกรณ์และสร้างแผนภาพการเคลื่อนที่สำหรับอธิบายการทำงาน ดังรูปที่ PP1 แสดงอุปกรณ์ลูกสูบ 2 ตัว (A, B) ที่ติดอุปกรณ์ตรวจจับตำแหน่งของก้านลูกสูบ (S1, S2, S3, S4) เครื่องหมาย ลบ(-) หมายถึงก้านลูกสูบอยู่ในตำแหน่งเข้าสู่สุด และเครื่องหมายบวก(+) หมายถึงก้านลูกสูบอยู่ในตำแหน่งออกสุด



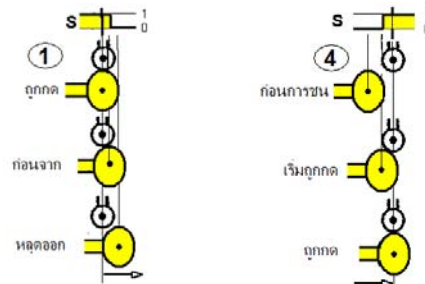
(ก) แผนภาพอุปกรณ์ทำงาน

(ข) แผนภาพการเคลื่อนที่

รูปที่ PP1 แสดงรูปแบบการทำงานของอุปกรณ์ทำงานและแผนภาพการเคลื่อนที่



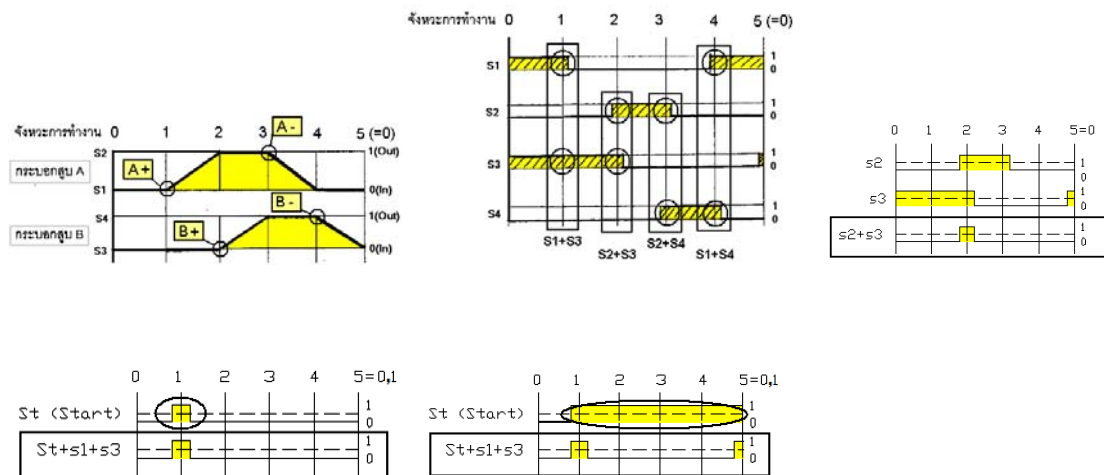
(ก) เริ่มเคลื่อนที่เข้าออก ในจังหวะที่ 1



(ข) ก่อนเคลื่อนที่เข้าชน ในจังหวะที่ 4

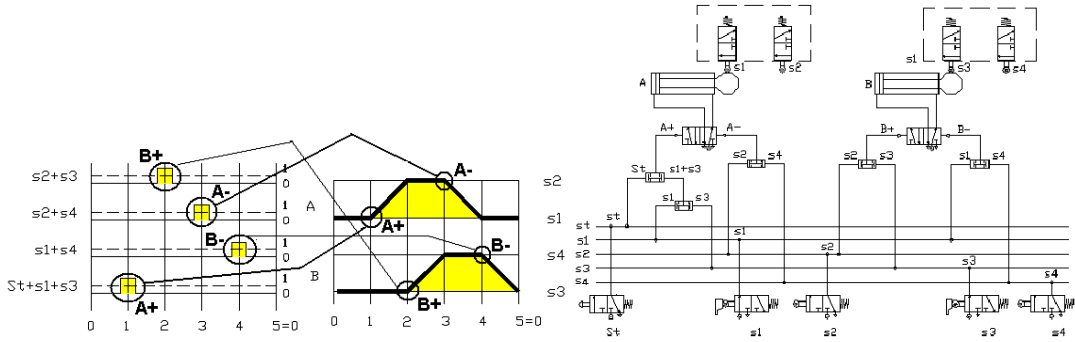
สวิทช์ยังคงถูกกดต่อเล็กน้อย

สวิทช์ถูกกดก่อนเล็กน้อย

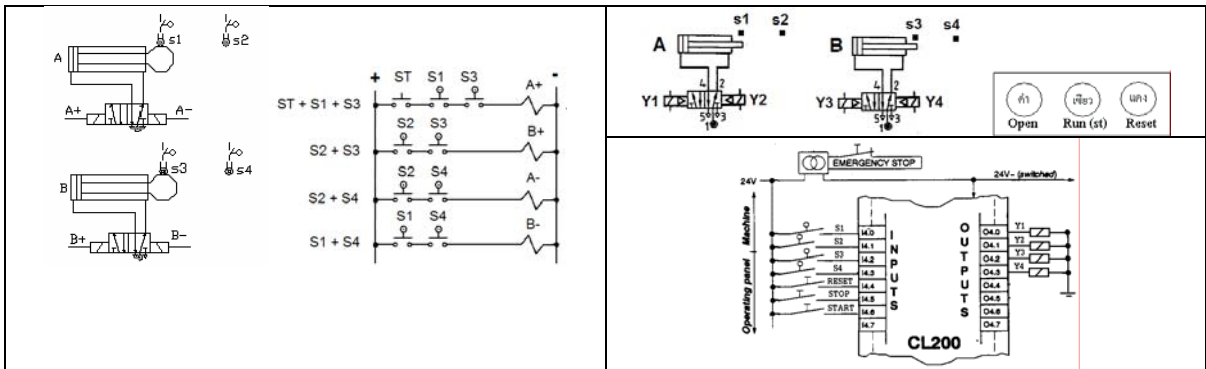


(ก) สัญญาณเริ่มต้นการทำงานแบบกึ่งอัตโนมัติ

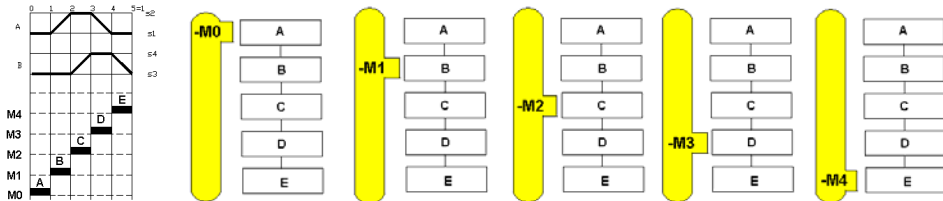
(ข) สัญญาณเริ่มต้นการทำงานแบบอัตโนมัติ



รูปที่ 5.1 แสดงการเลือกสัญญาณควบคุมกระบอกลูกสูบให้ทำงานในแต่ละจังหวะ



รูปที่ 5.2 แสดงวงจรนิวแมติกส์ที่สร้างจากการสัญญาณควบคุมจากรูปที่ 5.40



เลือกอุปกรณ์

วิเคราะห์สัญญาณตัวจับตำแหน่ง

วิเคราะห์จังหวัดการสั่งงาน

วิเคราะห์หาสัญญาณควบคุม

สร้างตาราง

สร้าง

จากรูปที่ 7.27 จากการวิเคราะห์ผลรวมสัญญาณตรวจจับ เพื่อสร้างสัญญาณควบคุมในบิตที่ 5 นั้นการทำงานยังไม่สมบูรณ์ เนื่องจากแม้การควบคุมจะสามารถทำงานตามที่ออกแบบได้ แต่อาจเกิดความผิดพลาดในการทำงานได้ ถ้าเกิดเหตุการณ์ที่สวิตช์บางตัวถูกกดโดยไม่ตั้งใจ ฉะนั้นเพื่อแก้ไขปัญหานั้นจะใช้วิธีการสร้างสัญญาณที่ลักษณะแบบขั้นบันได ดังแสดงในรูปที่ 7.27 เพื่อนำมารวมกับสัญญาณควบคุมดังแสดงในตารางที่ 7.19 เพื่อบังคับให้เกิดการทำงานตามลำดับและป้องกันการความผิดพลาดที่อาจเกิดขึ้น

รูปที่ 7.1 แสดงสัญญาณควบคุมและแผนภาพสัญญาณขั้นบันไดที่กำหนดขึ้น

จากรูปที่ 7.27 จะใช้ตัวมาคเกอร์จำนวน 5 ตัวได้แก่ M0, M1, M2, M3, M4 และ M5 โดยกำหนดให้มีค่าในแต่ละช่วงจังหวะการทำงาน ดังรูปที่ 7.28 แบ่งการทำงานออกเป็น 5 การทำงานย่อย ได้แก่ A B C D และ E โดยแต่ละงานย่อยจะเริ่มทำงานและมีการสั่งงานดังแสดงในตารางที่ 7.20 อธิบายการทำงานในงานย่อย A ดังนี้

1. สถานะสั่งงาน เกิดขึ้นเมื่อปุ่ม Open หรือ Reset ถูกกด โดยมีคำสั่งทำงานดังนี้

Reset กำหนดให้ M1 ถึง M5 ไม่มีค่าถาวร

Set กำหนดให้ M0, Y2 และ Y4 มีค่าถาวร

2. รอกการทำงาน จากการทำงานข้อที่ 1 ส่งผลให้กระบอบอกสูบ A และ B เคลื่อนที่เข้าสู่สุด

3. จบการทำงาน เมื่อ M0 S1 และ S3 มีค่า (กระบอบอกสูบ A และ B อยู่ในตำแหน่งเข้าสู่สุด) โดยมีคำสั่งทำงานดังนี้

Reset กำหนดให้ M0 ไม่มีค่าถาวร

Set กำหนดให้ M1 มีค่าถาวร

ตารางที่ 7.2 แสดงเงื่อนไขในการเปลี่ยนค่าสัญญาณขั้นบันไดในแต่ละลำดับ

ลำดับ	Start Condition	End Condition	Reset	Set (สั่งงาน)
A	Open or Reset	M0 + S1 + S3	M1, M2, M3, M4, Y1, Y3	M0, Y2 (A -), Y4 (B -)
B	M1 + St	M1 + S2 + S3	M0	M1
C	M2	M2 + S2 + S4	Y2	Y1 (A +)
D	M3	M3 + S1 + S4	M1	M2
E	M4	M4 + S1 + S3	Y4	Y3 (B +)
			M2	M3
			Y1	Y2 (A -)
			M3	M4
			Y3	Y4 (B -)
			M4	M1

จากตัวอย่างที่อธิบายในขั้นตอนงานย่อย A เมื่อสิ้นสุดขั้นตอนย่อย A แล้วอุปกรณ์ทุกตัวจะพร้อมเริ่มดำเนินการทำงานโดยจะมีเฉพาะค่า M1 เท่านั้นที่มีค่า และรอจนกว่าปุ่ม St ถูกกดก็จะเริ่มต้นการทำงานในขั้นตอนย่อย B เมื่อจบขั้นตอนย่อย B จะมีเฉพาะ M2 เท่านั้นที่มีค่าและจะเป็นสัญญาณในการสั่งให้ขั้นตอน C ทำงาน และเป็นลูกใช้จนจบการทำงานดังแสดงรายละเอียดในการเขียนเลดเดอร์โปรแกรมและอินสตรักชันวิธีโดยละเอียดดังตารางที่ 7.21

ตารางที่ 7.3 แสดงภาษาอินสตรักชันจากตารางที่ 7.19

ลำดับ	Instruction	List	อธิบาย	Ladder Diagram
A.1	A O	-OPEN -RESET	Start System Or Reset Error	

A.2	R R R R R S S S	-M1 -M2 -M3 -M4 -Y1 -Y3 -M0 -Y2 -Y4	A - B -	
	A A R S	-M0 -S1 -S3 -M0 -M1	Push Start Button (Run) M0 -> M1	
B.1	A A R S	-M1 -ST -Y2 -Y1	A +	
B.2	A A R S	-M1 -S2 -S3 -M1 -M2	M1 -> M2	
C.1	A R S	-M2 -Y4 -Y3	B +	
C.2	A A R S	-M2 -S2 -S4 -M2 -M3	M2 -> M3	
D.1	A R S	-M3 -Y1 -Y2	A -	
D.2	A A R S	-M3 -S1 -S4 -M3 -M4	M3 -> M4	
E.1	A R S	-M4 -Y3 -Y4	B -	
E.2	A A R S	-M4 -S1 -S3 -M4 -M1	M4 -> M1	

3. วิธีการดำเนินงานวิจัย

4. การวิเคราะห์ข้อมูลปริศนาคณิตศึกษา

1.

6. วิธีการในการประมวลผล

7. ขั้นตอนการใช้งานโปรแกรมที่พัฒนาขึ้น

4. บทสรุป

จากขั้นตอนที่กล่าวมาแล้วนั้น สามารถนำมาประยุกต์ใช้ทั้งในขั้นตอนการออกแบบระบบควบคุมด้วยระบบนิวแมติกส์ และระบบควบคุมนิวแมติกส์ด้วยหน้าสัมผัสรีเลย์ไฟฟ้า รวมทั้งการควบคุมด้วย PLC ซึ่งมีข้อแตกต่างจากการควบคุม

ทั้งสองระบบแรก คือ ต้องมีการสร้างสัญญาณขึ้นบันได เพื่อควบคุมลำดับการทำงานให้เป็นตามลำดับที่ถูกต้อง ซึ่งผลจากการนำไปใช้ในงานออกแบบระบบควบคุมที่ซับซ้อน และการสอนนักศึกษา นั้น พบว่าสามารถนำไปประยุกต์ใช้งานได้จริงตามที่ออกแบบ และง่ายต่อการเรียนรู้

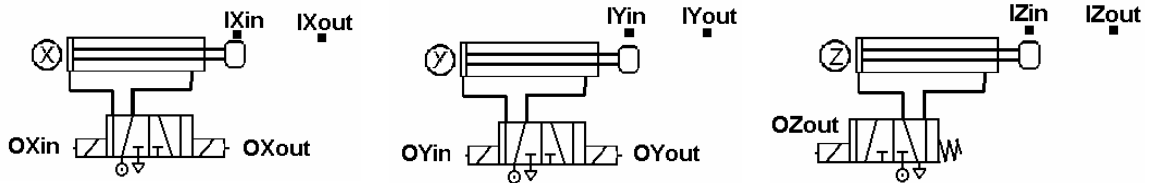
รูปที่ 5 แสดงผลต่างของจำนวนบรรจุภัณฑ์จากการคำนวณที่มากกว่าวิธีดั้งเดิม

5.4 ข้อเสนอแนะ

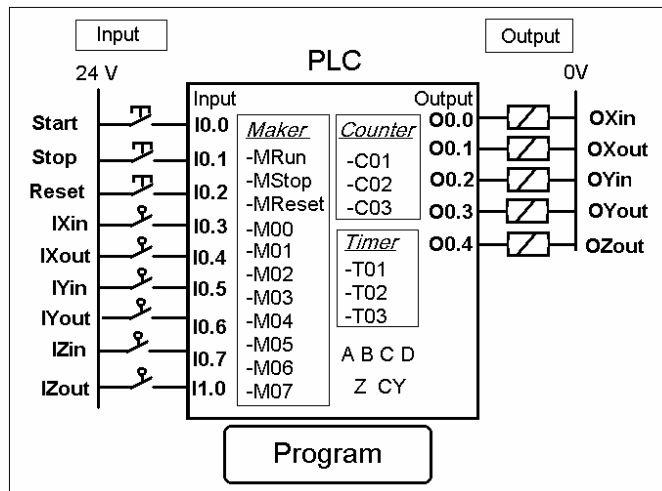
รายการอ้างอิง

[1] Dyckhoff H. 1990 European Journal of Operational Research. A. Typology of Cutting and Packing Problems 44

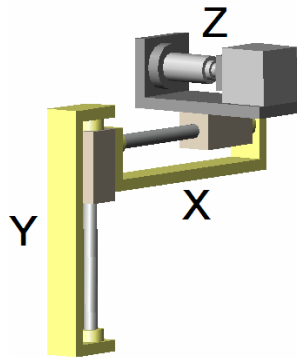
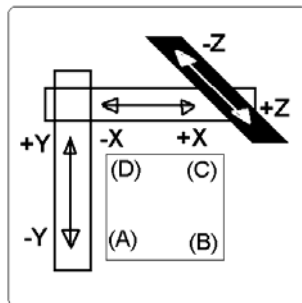
[2] โชคชัย ใจลานนท์. 2543. โครงการการบรรจุชิ้นส่วนเพื่อประโยชน์สูงสุด. วิทยานิพนธ์. ปริญญา มหาบัณฑิต. ภาควิชาวิศวกรรมอุตสาหกรรม บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.



รูปที่ 1 วงจรนิวแมติกส์ควบคุมระบบยกสูบสองทางด้วยวาล์ว 5/2



รูปที่ 2 แผนผังการต่ออุปกรณ์ควบคุมกับ PLC



ปั๊ม Reset
 X- Y- Z-
 ปั๊ม Start
 Y+ X+ Z+ Z- X-
 Y-

รูปที่ 3 อุปกรณ์ทำงานหยุดยกขึ้นงานเก็บเข้าชั้นวาง

แสดงเงื่อนไขในการเปลี่ยนค่าสัญญาณขึ้นบันไดในแต่ละลำดับ A ->D->C

ลำดับ	Start Condition	End Condition	Reset	Set (สั่งงาน)
0	Open or Reset		M1,M2,M3,M4,M5 OXOUT,OYOUT	M0 , OXIN,OYIN
		M0+IXIN+IYIN	M0	M1
1	M1+START		OXIN	OXOUT
		M1+IXOUT	M1	M2

2	M2		OYIN	OYOUT
		M2+IYOUT	M2	M3
3	M3			OZOUT
		M3+IZOUT	M3	M4
4	M4		OZOUT	
		M4+IZIN	M4	M5
5	M5		OXOUT,OYOUT	OXIN,OYIN
		M5+IXIN+IYIN	M5	M1