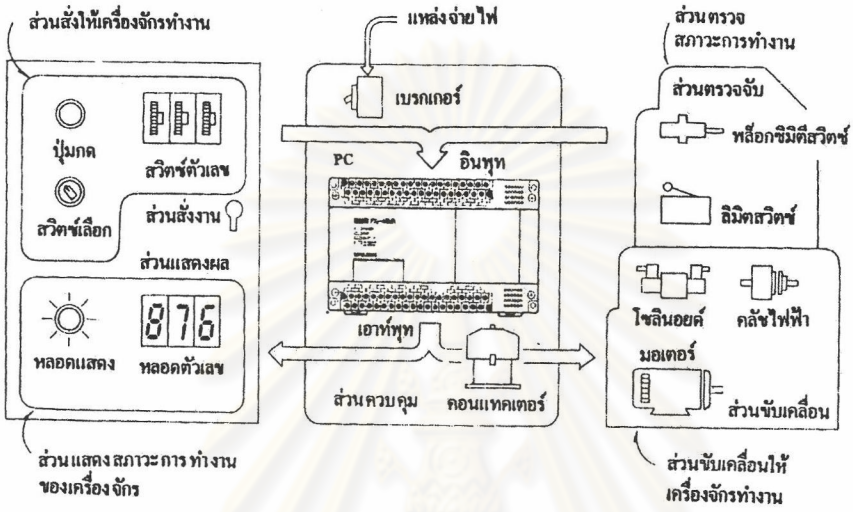
หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

พื้นฐานการทำงานของเครื่องควบคุมแบบลำดับที่โปรแกรมได้

2.1 ตัวควบคุมแบบลำดับที่โปรแกรมได้คืออะไร [2]



รูปที่ 2.1 การทำงานของ PLC

ตัวควบคุมแบบลำดับที่โปรแกรมได้หรือเรียกย่อๆ ว่า PLC (Programmable Logic Controller) เป็นอุปกรณ์ควบคุมการทำงานแบบลำดับ มีระบบการทำงานที่ควบคุมด้วยคอมพิวเตอร์ (Computer Control) มีไมโครโพรเซสเซอร์ (Microprocessor) เป็นหัวใจในการควบคุมการทำงาน PLC จะทำงานตามคำสั่งที่โปรแกรมเข้าไปโดยผ่านตัวป้อนโปรแกรม (Programming Console) หรือคอมพิวเตอร์ส่วนบุคคล โปรแกรมคำสั่งที่ใช้มีอยู่หลายระบบด้วยกัน แต่ที่นิยมใช้กันมากก็คือ โปรแกรมขั้นบันได (Ladder Diagram) ซึ่งเป็นแผนผังวงจรแบบดั้งเดิมที่ใช้กันมานานในระบบควบคุมการทำงานด้วยรีเลย์ ดังนั้นจึงทำให้ผู้คุ้นเคยกับระบบเดิมสามารถใช้งานตัว PLC ได้โดยไม่ยากนัก การป้อนโปรแกรมคำสั่งทำได้ง่ายโดยการแปลงวงจรแบบขั้นบันไดให้อยู่ในรูปของโปรแกรมที่ PLC สามารถเข้าใจ PLC จะอ่านสัญญาณจากอินพุต และจะให้สัญญาณตอบสนองไปยังเอาต์พุตตามการประมวลผลจากโปรแกรมควบคุมที่ป้อนให้กับมัน โปรแกรมควบคุมที่ป้อนให้กับมันจะเลียนแบบการทำงานของรีเลย์ ตัวตั้งเวลาและตัวนับ นอกเหนือไปจากนี้ PLC ยังมีคำสั่งการทำงานเกี่ยวกับการจัดการกับข้อมูล (Data Handling) ซึ่งเป็นส่วนที่ทำให้ PLC มีความสามารถเหนือกว่าวงจรรีเลย์ธรรมดามาก

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

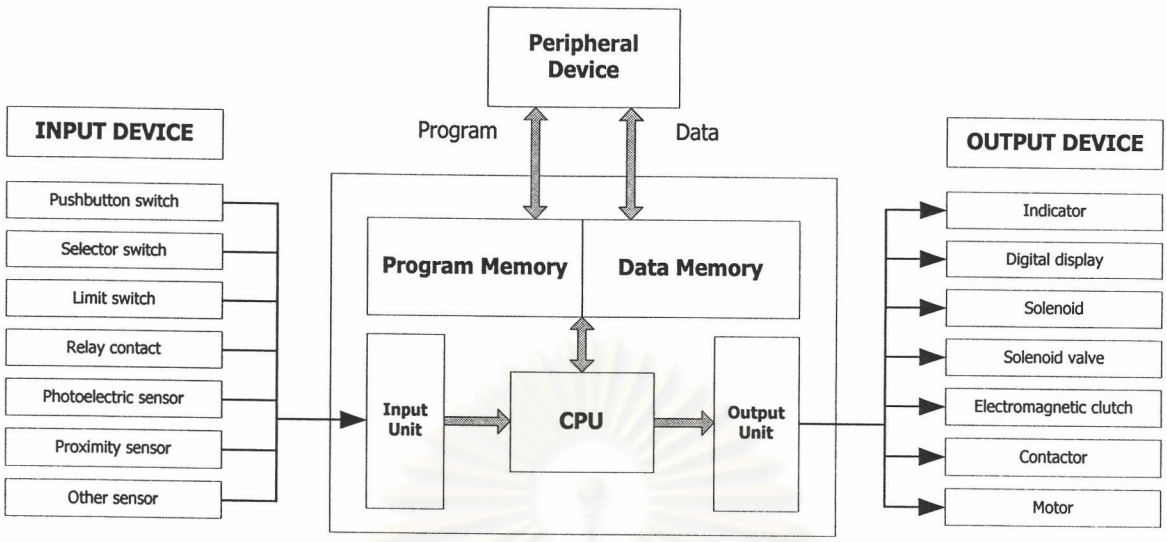
2.2 การทำงาน [3]

PLC สร้างขึ้นเพื่อใช้แทนวงจรรีเลย์ในการควบคุมแบบซีเควนซ์ (Sequence) หรือการทำงานตามลำดับแบบต่อเนื่อง เป็นการทำงานแบบอัตโนมัติทดแทนแบบเดิมที่มักใช้อุปกรณ์รีเลย์สร้างวงจรไฟฟ้าเพื่อควบคุมการทำงานของเครื่องจักรในโรงงานอุตสาหกรรม

ภายใน PLC ประกอบด้วยส่วนที่เรียกว่า CPU (Central Processing Unit) และหน่วยความจำที่ใช้เก็บข้อมูล (Memory) การทำงานของ PLC จะทำหน้าที่ในการควบคุมอุปกรณ์ภายนอกได้แก่ หลอดไฟ โซลินอยด์วาล์ว คลัตช์ไฟฟ้า และอื่นๆ โดยจะรับสัญญาณอินพุตจากอุปกรณ์ภายนอกซึ่งได้แก่ ลิมิตสวิตช์ สวิตช์ปุ่มกด สวิตช์เลือก หน้าสัมผัสของรีเลย์ โฟโตเซนเซอร์ พร็อกซิมิตี้เซนเซอร์ และเซนเซอร์อื่นๆ

การทำงานของ PLC จะทำงานโดยการรับสัญญาณจากอินพุตภายนอกเข้ามาทางหน่วยอินพุต (Input Unit) แล้วประมวลผลผ่านตัวไมโครคอมพิวเตอร์ ซึ่งจะทำงานตามลำดับขั้นที่เขียนไว้ในหน่วยความจำรวมเรียกว่าส่วนควบคุม จากนั้นจึงส่งผลลัพธ์ที่ได้ออกไปที่หน่วยเอาต์พุต (Output Unit) เพื่อไปควบคุมอุปกรณ์เอาต์พุตภายนอกอีกทีหนึ่ง ดังนั้นเราจึงไม่จำเป็นต้องมีความรู้มากมายก็สามารถใช้งาน PLC ได้ การนำเอา PLC ไปใช้ก็สะดวก เพียงแค่ออกแบบวงจรในส่วนที่เป็นอินพุตและเอาต์พุตของ PLC เท่านั้น การเดินสายของวงจรก็ไม่ยุ่งยากซับซ้อน ไม่เหมือนการออกแบบด้วยวงจรรีเลย์ที่ต้องมีการเดินสายจากหน้าหน้าสัมผัสหนึ่งไปยังอีกหน้าหน้าสัมผัสหนึ่ง ถ้าหากมีการใช้หน้าหน้าสัมผัสมากกว่าจำนวนที่รีเลย์มีก็จำเป็นจะต้องเพิ่มจำนวนรีเลย์เข้ามาในวงจร ซึ่งจะสิ้นเปลืองทั้งในส่วนของอุปกรณ์ที่เพิ่มเข้ามา กระแสไฟฟ้าที่ใช้มากขึ้น พื้นที่ที่ต้องการในการติดตั้งเพิ่มขึ้น และการเดินสายของวงจรที่ยุ่งยากยิ่งขึ้น นอกจากนี้ถ้าหากวงจรมีการใช้ตัวนับ หรือตัวตั้งเวลาก็ต้องมีการเพิ่มอุปกรณ์เหล่านี้ลงไปใน ตู้ควบคุมอีก ทำให้มีค่าใช้จ่ายเพิ่ม แต่สำหรับวงจรที่ใช้ PLC ในการควบคุมการทำงานแล้ว ภายในตัว PLC เองมีอุปกรณ์ตัวนับ ตัวตั้งเวลา ให้เรียกใช้งานได้ทันทีไม่ต้องมีฮาร์ดแวร์ให้ ยุ่งยาก จำนวนของตัวนับ ตัวตั้งเวลามากหรือน้อยขึ้นอยู่กับ PLC ของแต่ละบริษัทที่กำหนดให้ใช้งานได้ และถ้ามีการกำหนดเงื่อนไขในการทำงาน ของวงจรใหม่ก็ไม่ต้องเสียเวลาในการเดินสายวงจรใหม่ เพียงแต่เขียนโปรแกรมใหม่เท่านั้น

2.3 โครงสร้างของ PLC [4]



รูปที่ 2.2 โครงสร้างการทำงานภายใน PLC

1. **ซีพียู** (Central Processing Unit) ทำหน้าที่เป็นหน่วยประมวลผลกลาง ทำงานประสานงานกับทุกหน่วย
   - อ่านข้อมูลจากอินพุต แล้วนำข้อมูลมาประมวลผลตามโปรแกรมควบคุมที่เก็บไว้ในหน่วยความจำ ผลลัพธ์ที่ได้จากการประมวลผลจะถูกส่งออกไปที่เอาต์พุตเพื่อควบคุมอุปกรณ์ภายนอก
   - ทำหน้าที่รับส่งข้อมูลกับ Peripheral Device
   - ตรวจเช็คโปรแกรมควบคุมที่ป้อนเข้าไปเก็บในหน่วยความจำว่าถูกต้องหรือไม่ ถ้าไม่ถูกต้องก็จะแจ้งให้ผู้ใช้ทราบเพื่อทำการแก้ไขให้ถูกต้อง

2. **หน่วยความจำข้อมูล** (Data Memory) เป็นหน่วยความจำที่ใช้เก็บข้อมูลที่เป็นสภาวะการทำงานของอุปกรณ์ภายใน เช่น สภาวะการทำงานของรีเลย์ สภาวะของอินพุตและเอาต์พุตว่าทำงานหรือไม่อย่างไร นอกจากนี้ยังใช้เก็บสถานะการทำงานของตัวตั้งเวลา และตัวนับทุกตัวอีกด้วย

3. **หน่วยความจำโปรแกรม** (Program Memory) เป็นหน่วยความจำที่ใช้เก็บโปรแกรมควบคุมการทำงานของผู้ใช้ จะแปลงมาจากวงจรควบคุมแบบขั้นบันได (Ladder Diagram) แล้วป้อนเข้ามาเก็บในหน่วยความจำโดยผ่านตัวป้อนโปรแกรม (Programming Console) หรือจากเครื่องคอมพิวเตอร์ ซีพียูของ PLC จะอ่าน

โปรแกรมควบคุมในหน่วยความจำทีละคำสั่ง แล้วทำงานตามคำสั่งที่อ่านทันทีและจะทำคำสั่งถัดไปจนจบโปรแกรม (คำสั่งสุดท้าย) แล้วจะเริ่มต้นทำคำสั่งใหม่วนเวียนเช่นนี้ตลอดไป หน่วยความจำนี้ยิ่งมีมากเท่าไรก็สามารถบรรจุโปรแกรมควบคุมที่มีขนาดใหญ่และซับซ้อนได้มากยิ่งขึ้น

4. **ภาคอินพุต (Input Unit)** ทำหน้าที่รับสัญญาณจากสวิตช์และเซนเซอร์ (Sensor) ต่างๆ ในระบบแล้วแปลงให้เป็นระดับสัญญาณไฟที่เหมาะสมที่ PLC สามารถรับได้ สัญญาณไฟที่ป้อนเข้ามายังอินพุตอาจเป็นสัญญาณไฟ AC หรือ DC ก็ได้

   อินพุตอาจแบ่งออกได้เป็น 2 ชนิด คือ
   อินพุตชนิด AC (AC Input Type) มักใช้เป็นแบบแยกสัญญาณ (Isolate) ออกจากซีพียู เหมาะสำหรับต่อกับหน้าสัมผัส (Contact Input) ที่มาจากกระบวนการผลิต (Process) เช่น ลิมิตสวิตช์และคอนแทคเตอร์ เป็นต้น
   อินพุตชนิด DC (DC Input Type) มีทั้งแบบกระแสเข้า (Current Sink Input) และกระแสออก (Current Source Input) มาจากหน้าสัมผัสของสวิตช์ เซนเซอร์และทรานซิสเตอร์ขาออก

5. **ภาคเอาต์พุต (Output Unit)** รับสัญญาณจากซีพียูแล้วทำหน้าที่ขยายสัญญาณออกให้มีขนาดใหญ่พอที่จะขับอุปกรณ์ภายนอก เช่น คอนแทคเตอร์และโซลินอยล์วาลว์ เป็นต้น นอกจากนี้ยังทำหน้าที่แยกสัญญาณภายในและสัญญาณภายนอกออกจากกัน เพื่อป้องกันระดับสัญญาณไฟสูงจากภายนอกเข้ามาทำความเสียหายแก่ซีพียู โดยปกติจะมีความสามารถในการขับโหลดด้วยกระแสประมาณ 2-5 แอมแปร์

   ภาคเอาต์พุตมีด้วยกันหลายชนิด การใช้งานจะต้องพิจารณาเลือกใช้ให้เหมาะสมคือ
   1. เอาต์พุตชนิด AC (AC Output) มักจะเป็นแบบแยกสัญญาณ (Isolate) มีทั้งแบบหน้าสัมผัส (Contact Output) หรือแบบไตรแอค (Triac Output) สามารถใช้งานได้ทนทานเหมาะสำหรับโหลดที่มีการเปิดปิดบ่อยๆ
   2. เอาต์พุตชนิด DC (DC Output) มีทั้งแบบแยกสัญญาณ (Isolate) และแบบไม่แยกสัญญาณ (Non-Isolate)
      - แบบหน้าสัมผัส (Contact Output)
      - แบบทรานซิสเตอร์ (Transistor Output)

6. PERIPHERAL DEVICE เป็นอุปกรณ์แบบต่างๆ ที่ใช้อำนวยความสะดวกในการพัฒนาโปรแกรม เช่น ตัวป้อนโปรแกรม จอแสดงผล และเครื่องพิมพ์ เป็นต้น

2.4 การเขียนโปรแกรม PLC

การเขียนโปรแกรมสั่งงาน PLC สามารถทำได้โดยการเขียนโปรแกรมให้อยู่ในรูปแบบภาษาที่ PLC เข้าใจ ซึ่งใน PLC แต่ละรุ่นแต่ละบริษัทจะมีความแตกต่างกันไปตามพื้นฐานการวางรูปแบบของคำสั่ง แต่โดยหลักการจะอาศัยพื้นฐานการทำงานของวงจรตรรกกะ (Logic) เหมือนกัน ซึ่งได้แก่คำสั่งพวก AND, OR เป็นหลัก นอกจากนั้นก็จะมีคำสั่งที่เพิ่มเติมเข้ามาโดยที่คำสั่งที่แต่ละบริษัทสร้างขึ้นก็เพื่อให้เหมาะสมกับลักษณะของงานและง่ายต่อการเขียนคำสั่ง โดยผู้ใช้ไม่จำเป็นที่จะต้องมีความรู้มากนักก็สามารถใช้งานได้ คำสั่งหรือภาษาที่ใช้ในการสั่งงานให้ PLC ทำงานนั้นในอดีตค่อนข้างจะยุ่งยาก และต้องมีพื้นฐานมากพอสมควรจึงจะเข้าใจ เนื่องจาก PLC นั้นเริ่มต้นพัฒนามาจากไมโครคอนโทรลเลอร์รูปแบบของคำสั่งจึงอยู่ในรูปของภาษาเครื่อง (Machine Code) หรือเราเรียกอีกอย่างหนึ่งว่า ภาษานิวโมนิค (Mnemonic Code) มีรูปแบบคล้ายกับภาษาแอสแซมบลี้ (Assembly)

การป้อนโปรแกรมให้กับ PLC ในรูปแบบของภาษาเครื่องจะใช้อุปกรณ์ที่เราเรียกว่า โปรแกรมคอนโซล (Program Console) โปรแกรมมือถือ (Handheld Programmer) หรืออาจจะเป็นเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal Computer) การป้อนโปรแกรมก็คือการป้อนคำสั่งเพื่อสั่งงานให้ PLC ทำงานตามลำดับขั้น (Step) ซึ่งก็ยุ่งยากพอควรหากไม่สามารถเข้าใจคำสั่งอย่างถูกต้อง ปัจจุบันได้มีการปรับปรุงพัฒนาการป้อนโปรแกรมให้ง่ายขึ้นด้วยรูปแบบของภาษาที่เราเรียกว่า ภาษาแลดเดอร์ (Ladder) หรือภาษาแบบขั้นบันไดเป็นการเขียนคำสั่งในลักษณะของสัญลักษณ์ของหน้าสัมผัสแทนอินพุตหรือหน้าหน้าสัมผัสของรีเลย์ นอกจากนี้ยังมีคำสั่งที่ใช้แทนตัวนับ ตัวตั้งเวลาและคำสั่งอื่นๆ ที่มีเพิ่มเติมสำหรับการทำงานที่ซับซ้อนขึ้น การทำงานที่มีเงื่อนไข การทำงานแบบรอทำงานทีละขั้น การทำงานแบบมีโปรแกรมย่อย และการทำงานที่มีการคำนวณ การประมวลผล การสั่งงานในลักษณะต่างๆ กันเหล่านี้ช่วยให้ผู้ใช้ทำงานได้สะดวกขึ้น แต่ตัวของภาษาที่ใช้สั่งงานให้กับ PLC จริงๆ ก็คือภาษาเครื่องไม่ใช่ภาษาแบบขั้นบันได ตัวคอมไพเลอร์ (Compiler) ทำหน้าที่แปลงความหมายจากภาษาแบบขั้นบันไดที่เราเขียนให้เป็นภาษาเครื่องที่ PLC เข้าใจอีกทีหนึ่ง

Step No.	Instruction	Operand
001	LD	0000
002	AND	0001
003	OR	1000
004	AND	0002
005	OUT	0500
006	END	

ตารางที่ 2.1 โปรแกรมภาษาของ PLC

โปรแกรมของ PLC ประกอบไปด้วยชุดคำสั่งจำนวนมากทำงานเรียงกันตามลำดับหมายเลขขั้น (Step Numbers) หรือลำดับการทำงานของคำสั่งในโปรแกรมดังแสดงในตารางที่ 2.1 โดยแต่ละคำสั่งจะมีหมายเลขของอินพุตหรือเอาต์พุตกำกับ ซึ่งเรียกว่าโอเปอร์แลน (Operand) ในการกำหนดหมายเลขของอินพุตหรือเอาต์พุตจะขึ้นอยู่กับ PLC แต่ละรุ่นที่นำมาต่อใช้งานรวมถึงการต่อส่วนขยายของอินพุตหรือเอาต์พุตให้กับ PLC ในการใช้งานด้วย

2.5 ระบบการโปรแกรม (Programming System) [1]

PLC ทำงานได้ด้วยโปรแกรมที่ถูกป้อนโดยผู้ใช้ โดยภาษาที่สามารถป้อนให้กับ PLC มีด้วยกันหลายภาษาตามการออกแบบของผู้ผลิต แต่ทุกภาษาต้องสามารถเปลี่ยนให้เป็นภาษาแบบนิวโมนิคได้เพื่อแปลโปรแกรมให้เป็นภาษาเครื่องที่ PLC เข้าใจได้ต่อไป

1. Boolean Logic Equation

$$I1 \;\&\; (\sim I2 \;\&\; I3 + \sim I4) = O2$$

รูปที่ 2.3 สมการบูลีน

รูปที่ 2.3 แสดงตัวอย่างของสมการบูลีน ใช้เขียนขึ้นเพื่อแสดงการคำนวณค่าของอินพุต 4 ตัวจากสมการ แล้วส่งผลลัพธ์ไปยังเอาต์พุต ภาษานี้มีประโยชน์เพื่อนำมาใช้กระทำทางตรรกะแบบง่ายๆ ได้มีการพัฒนาสมการบูลีน (Boolean Language) โดยมีความสามารถในการกระทำฟังก์ชันที่สลับซับซ้อน เพื่อนำมาใช้กับ PLC เช่นมีการเพิ่มให้มีตัวตั้งเวลาและตัวนับ เป็นต้น

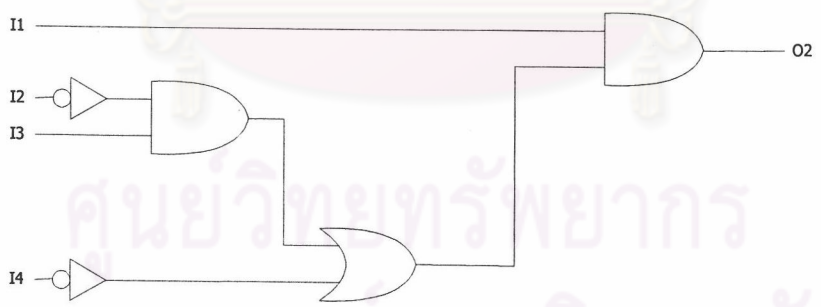
2. The Mnemonic Language

```
LD    I1
LDI   I2
AND   I3
LDI   I4
ORB
OUT   O2
```

รูปที่ 2.4 ภาษานิวโมนิค

รูปที่ 2.4 แสดงตัวอย่างของภาษานิวโมนิค (Mnemonic) ที่นำไปใช้เขียนเป็นโปรแกรมควบคุม โดยจะเป็นรหัสคำสั่งที่ง่ายในการแปลงให้เป็นภาษาเครื่อง และภาษานี้สามารถเขียนฟังก์ชันเพิ่มเติมเพื่อนำไปควบคุมได้อีก แต่จะต้องมีความรู้เกี่ยวกับการทำงานภายในเครื่องเป็นอย่างดี และการอ่านผลลัพธ์ของโปรแกรมทำได้ค่อนข้างยาก

3. The Logic Diagram Language

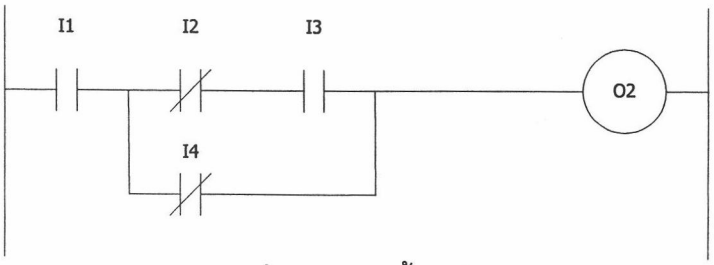


รูปที่ 2.5 ภาษา Logic Diagram

ภาษาที่เขียนเป็นวงจรโดยใช้เกตต่างๆ นำมาต่อกันเป็นวงจรควบคุมที่ต้องการ ดังแสดงในรูปที่ 2.5 ภาษานี้เป็นภาษาที่ง่ายในการเข้าใจถ้าวงจรควบคุมที่ต่อเป็นแบบง่ายๆ แต่ความสามารถในการขยายให้ทำฟังก์ชันที่ซับซ้อนทำได้ค่อนข้างยาก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4. Ladder Diagram Language



รูปที่ 2.6 ภาษาขั้นบันได

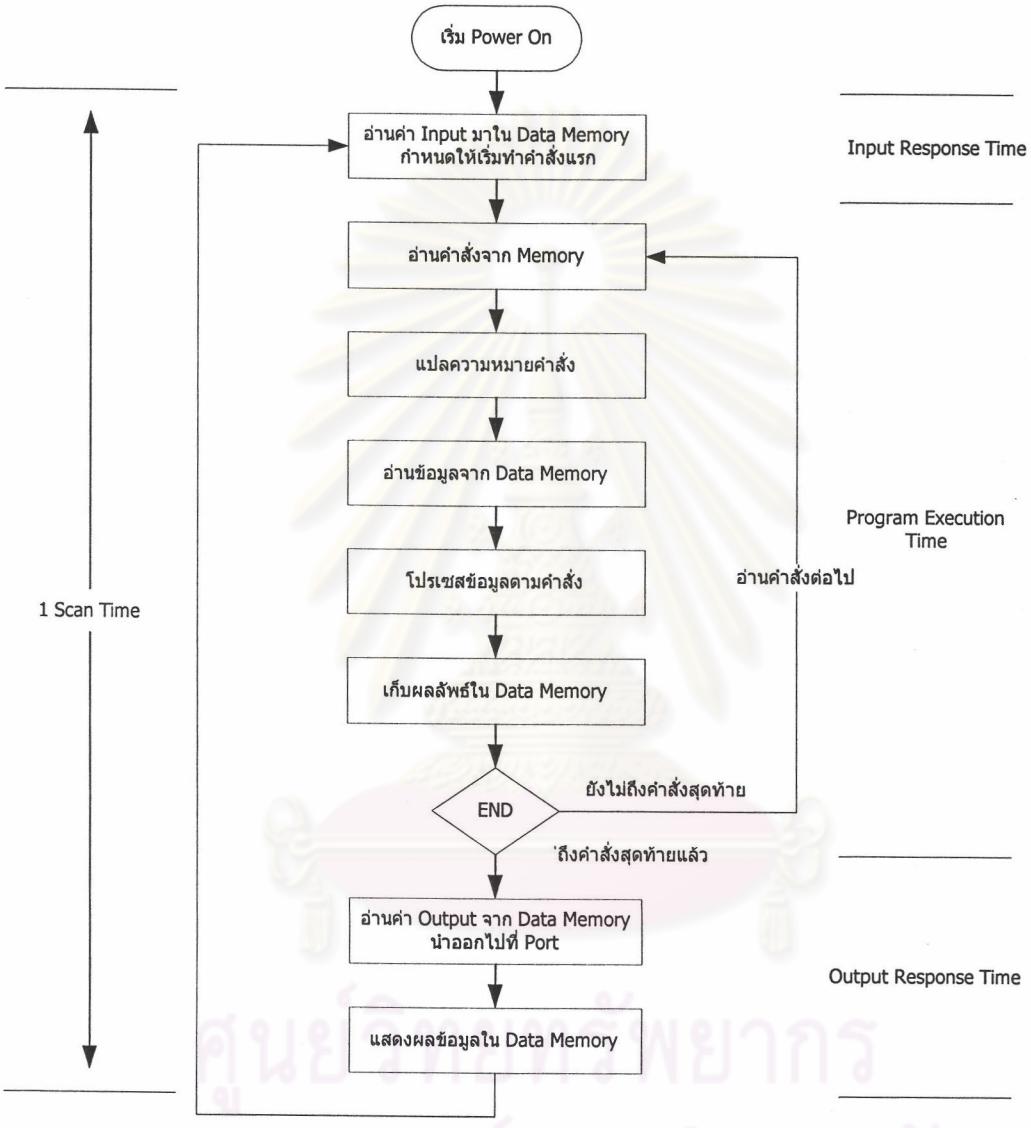
รูปที่ 2.6 แสดงตัวอย่างของโปรแกรมภาษาขั้นบันได (Ladder Diagram) ซึ่งเหมือนกับวงจรที่ใช้ควบคุมระบบรีเลย์ ภาษาโปรแกรมขั้นบันไดเป็นที่นิยมใช้ในการโปรแกรม PLC อย่างมากเพราะ

- การแปลงภาษาจากวงจรที่ใช้ควบคุมระบบรีเลย์ให้มาเป็นภาษาเพื่อป้อนให้ PLC ทำได้ง่าย
- การตรวจซ่อม เพื่อหาจุดผิดพลาดของวงจรทำได้ง่าย
- เป็นภาษาที่ง่ายต่อการอ่าน และทำความเข้าใจ
- สามารถขยายให้ทำฟังก์ชันที่สลับซับซ้อนได้ง่าย โดยการเพิ่ม Function Block ลงไปใน Ladder Diagram

2.6 วงรอบการทำงานและ Scan Time

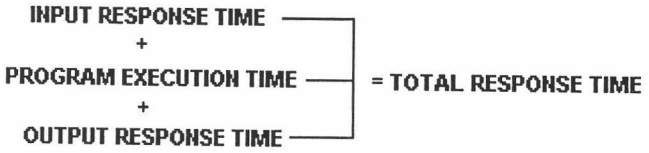
PLC มีการลักษณะการทำงานเป็นวงรอบดังแสดงในรูปที่ 2.7 เริ่มจากเมื่อเปิดเครื่องมีไฟเลี้ยงเข้ามา PLC จะเริ่มอ่านค่าอินพุตจาก Input Module มาเก็บไว้ในหน่วยความจำข้อมูล จากนั้นก็ทำงานตามโปรแกรมขั้นบันไดที่ผู้ใช้กำหนด โดยทำงานจากคำสั่งแรก อ่านข้อมูลโปรแกรมจากหน่วยความจำที่เก็บโปรแกรมขั้นบันได จากนั้นทำการแปลงความหมาย และอ่านข้อมูลจากหน่วยความจำข้อมูลที่คำสั่งนั้นๆ อ้างถึง แล้วทำการดำเนินการตามคำสั่ง ผลลัพธ์ที่ได้นำกลับมาเก็บหน่วยความจำข้อมูล แล้วตรวจสอบว่าได้ทำงานตามคำสั่งโปรแกรมขั้นบันไดจนถึงคำสั่งสุดท้ายหรือยัง ถ้ายังก็ทำคำสั่งถัดไป แต่ถ้าถึงคำสั่งสุดท้ายแล้วทำการอ่านค่าเอาต์พุตจากหน่วยความจำข้อมูลส่งออกไปยัง Output Unit และจากนั้นอาจมีการส่งข้อมูลสถานะภายในหน่วยความจำข้อมูลไปแสดงผลที่เครื่องคอมพิวเตอร์ จึงถือว่า PLC ได้ทำงานครบหนึ่งรอบการทำงาน หรือมีชื่อเรียกว่า 1 Scan Time แล้วจึงเริ่มทำงานในวงรอบต่อไป

ดังนั้น 1 Scan Time ก็คือ เวลาทั้งหมดที่ใช้ในหนึ่งรอบการทำงาน ตั้งแต่การอ่านค่าอินพุตจนไปถึงได้ค่าเอาต์พุตและแสดงผลออกมา ซึ่งอาจมีหน่วยเป็น มิลลิวินาทีหรือไมโครวินาทีขึ้นอยู่กับความเร็วของ PLC นั้นๆ



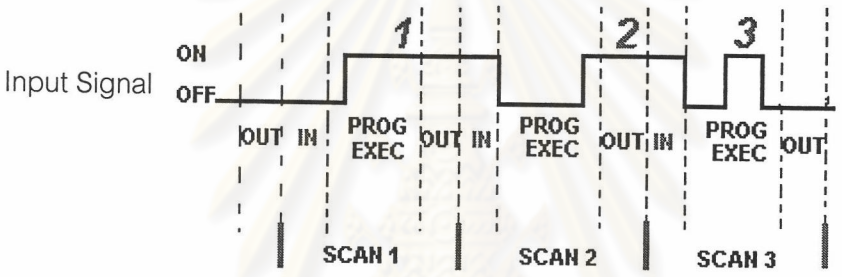
รูปที่ 2.7 โฟล์วชาร์ตวงรอบการทำงานและ Scan Time

2.7 ความเร็วของ PLC ที่มีผลต่อการควบคุม [5]



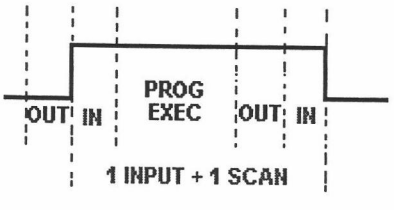
รูปที่ 2.8 การหาเวลาการตอบสนองทั้งหมด

เวลาการตอบสนอง (Response time) คือ เวลาที่ใช้ในการอ่านอินพุตรวมกับเวลาที่ใช้ในการทำตามคำสั่งขั้นบันไดทั้งหมดรวมกับเวลาในการเปลี่ยนให้ค่าเอาต์พุตออกมา

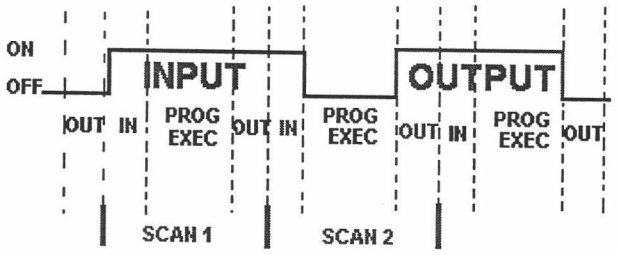


รูปที่ 2.9 Scan Time และสัญญาณอินพุตที่เข้ามา

ความเร็วของ PLC มีผลต่อการควบคุมจากรูปที่ 2.9 เวลาการตอบสนอง (Response time) ก็คือเวลาหนึ่งรอบการทำงานนั้นเอง จากรูปที่ 2.9 จะเห็นว่า Scan time 1 ก็ยังไม่สามารถจับอินพุตที่เปลี่ยนแปลงได้ทัน ต้องรอจนถึง Scan time 2 ส่วน Scan time 3 จะไม่เห็นช่วง off แต่จะเห็นช่วง On เลย ทำให้ไม่สามารถจับ Pulse ที่สามได้ เราสามารถแก้ไขปัญหานี้ได้สองวิธีคือ หนึ่งให้สัญญาณอินพุต Pulse ยาวขึ้น (จำกัดความถี่สัญญาณอินพุต) ดังรูปที่ 2.10 หรือสองคือลดเวลาการตอบสนอง Scan time ลง นั้นก็คือการเพิ่มความเร็วของ PLC นั้นเอง



รูปที่ 2.10 Pulse แคบที่สุด PLC รับได้ เท่ากับ 1 Input + 1 Scan Time



รูปที่ 2.11 กรณีที่แย่ที่สุดของการ Delay

รูปที่ 2.11 แสดงกรณีที่แย่ที่สุดเมื่อมีการเปลี่ยนอินพุตแล้วให้เอาต์พุตหน่วงเวลามากที่สุดคือ 2 Scan time – 1 input delay time

ปกติแล้วกำหนดให้เวลาในการทำงาน 1 รอบ (Scan time) ควรมีค่าน้อยกว่า 50 ไมโครวินาที ถ้าเวลานี้ช้าเกินไปอาจทำให้ซีพียูไม่สามารถรับรู้การเปลี่ยนแปลงของสัญญาณอินพุตได้ทัน ในการทำงาน 1 รอบจะเห็นว่า INPUT/OUTPUT Response time จะคงที่ ดังนั้นถ้า PLC ไม่เร็วพอจะทำให้มีคำสั่งขั้นบันไดได้ไม่มาก

2.8 การอ่านเขียนอินพุต/เอาต์พุต [4]

การอ่านและเขียนอินพุต/เอาต์พุตของเครื่อง PLC สามารถแบ่งออกได้เป็น 2 วิธี คือ วิธี Continuous updating และวิธี Mass Input/Output Copy ทั้ง 2 วิธีมีลักษณะที่แตกต่างกันคือ

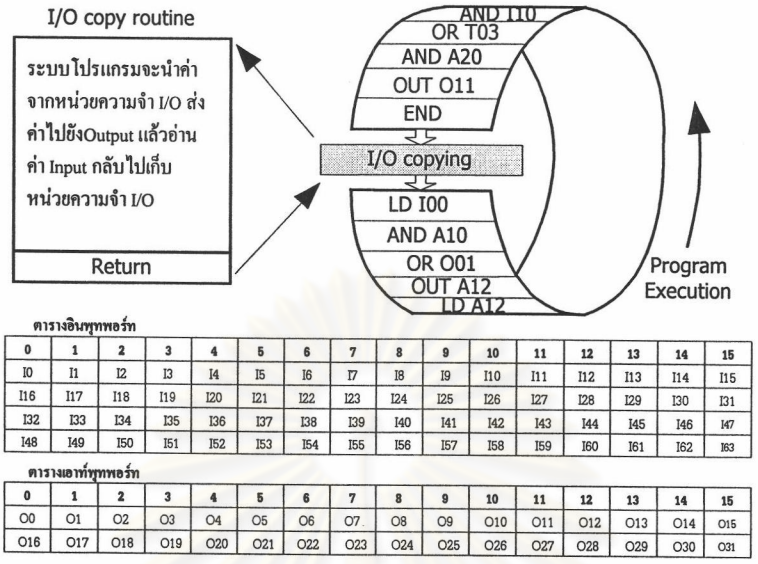
1. **วิธี** Continuous Updating

EXECUTE FIRST INSTRUCTION	SCAN I/O	EXECUTE NEXT INSTRUCTION	SCAN I/O	EXECUTE NEXT INSTRUCTION	SCAN I/O

รูปที่ 2.12 การอ่านเขียนแบบ Continuous Updating

ซีพียูจะอ่านและเขียนอินพุต/เอาต์พุต เมื่อกำลังทำงานคำสั่งนั้นในโปรแกรมขั้นบันไดโดยตรง ซึ่งถ้าเป็นอินพุตก็จะได้ค่าจริงของข้อมูลที่มี Delay Time ตามอินพุตพอร์ตนั้น ถ้าเป็นเอาต์พุตซีพียูก็จะเขียนเอาต์พุตพอร์ตโดยตรง การทำงานของวิธีนี้มีลักษณะดังแสดงในรูปที่ 2.12

2. **วิธี** Mass Input/Output Copy



ตารางอินพุทพอร์ท

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I0	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13	I14	I15
I16	I17	I18	I19	I20	I21	I22	I23	I24	I25	I26	I27	I28	I29	I30	I31
I32	I33	I34	I35	I36	I37	I38	I39	I40	I41	I42	I43	I44	I45	I46	I47
I48	I49	I50	I51	I52	I53	I54	I55	I56	I57	I58	I59	I60	I61	I62	I63

ตารางเอาท์พุทพอร์ท

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
O0	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	O11	O12	O13	O14	O15
O16	O17	O18	O19	O20	O21	O22	O23	O24	O25	O26	O27	O28	O29	O30	O31

รูปที่ 2.13 การอ่านเขียนแบบ Mass input/output copy และตารางในการเก็บค่าอินพุต/เอาต์พุต

วิธี Mass input/output copy ตามรูปที่ 2.13 สร้างตารางในการเก็บสถานะของอินพุต/เอาต์พุตในหน่วยความจำ การทำงานของซีพียูตามคำสั่งโปรแกรมขั้นบันไดก็จะติดต่อกับตารางหน่วยความจำที่เก็บข้อมูลแทน เมื่อทำงานครบหนึ่งรอบทำงาน (Scan time) ก็จะอ่านข้อมูลจากอินพุตมาเก็บในตาราง และนำค่าเอาต์พุตที่เก็บในตารางส่งไปยังเอาต์พุตพอร์ต ซึ่งจะมีการคงสถานะ (Latch) ของข้อมูลไว้จนกว่าจะมีการเขียนครั้งใหม่ วิธีนี้จะทำให้ค่าของอินพุต/เอาต์พุตเหมือนกันตลอดการทำงานหนึ่งรอบการทำงาน (Scan time)

ถ้าเครื่อง PLC ที่มีจำนวนอินพุต/เอาต์พุตมาก หรือมีจำนวนขั้นของโปรแกรมขั้นบันไดยาว สถานะของอินพุตขณะที่ทำงานต้นโปรแกรมขั้นบันไดและสถานะของอินพุตขณะที่ทำงานท้ายโปรแกรมบันไดอาจมีค่าแตกต่างกัน ซึ่งอาจมีผลทำให้การทำงานของโปรแกรมควบคุมไม่ถูกต้องถ้าใช้วิธี Continous updating และเพื่อแก้ปัญหานี้ PLC ในวิทยานิพนธ์นี้จึงเลือกใช้วิธี Mass input/output copy ในการอ่านเขียนอินพุต/เอาต์พุต

2.9 การวิจัยและพัฒนา PLC ในประเทศไทย

การวิจัยและพัฒนาสร้าง PLC ในประเทศเริ่มต้นมานานแล้ว ในปี 2532 [1] ได้มีการนำไมโครโพรเซสเซอร์เบอร์ 8085 มาสร้างเป็นหน่วยประมวลผลกลาง PLC ที่ความถี่ในการทำงาน 6 MHz มีจำนวนอินพุต/เอาต์พุต จำนวน 32 จุด ภาษาที่ใช้ในควบคุมคือโปรแกรมภาษาขั้นบันได (Ladder Diagram) สามารถโปรแกรมได้สูงสุด 1000 ขั้น (Step) มีความเร็วเฉลี่ยในการทำงานของคำสั่งพื้นฐาน 15.6 ไมโครวินาทีต่อขั้น ผลที่ได้จากการนำไปทดสอบควบคุมเครื่องจักรจริงก็เป็นที่น่าพอใจ

ถัดมาในปี 2533 [3] ได้มีการพัฒนา PLC ให้ดีขึ้น โดยเพิ่มคำสั่งในด้านจัดการข้อมูล (Data Handling) ใช้ไมโครโพรเซสเซอร์เบอร์ Z80CTC เป็นหน่วยประมวลผลกลางของ PLC ความถี่ในการทำงาน 4 MHz มีจำนวนอินพุต/เอาต์พุต จำนวน 512 จุด ภาษาที่ใช้ในควบคุมคือโปรแกรมภาษาขั้นบันได (Ladder Diagram) สามารถโปรแกรมได้สูงสุด 4000 ขั้น (Step) มีความเร็วเฉลี่ยในการทำงานของคำสั่งพื้นฐาน 5.7 ไมโครวินาทีต่อขั้น ผลที่ได้จากการนำไปทดสอบควบคุมเครื่องจักรจริงก็เป็นที่น่าพอใจเช่นกัน

จากนั้นก็ได้มีการนำเสนอบทความในการประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 18 เมื่อปี 2538 [6] เรื่องเครื่องควบคุมแบบตรรกที่โปรแกรมได้ แต่ก็เป็นเพียงการศึกษาโครงสร้างและการทำงานภายในของ PLC OMRON ไม่ได้มีการพัฒนาสร้างเครื่องต้นจริง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย