หอสมุดกลาง จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ความรู้เบื้องต้นเกี่ยวกับตัวควบคุมแบบลำดับที่โปรแกรมได้

2.1 ตัวควบคุมแบบลำดับที่โปรแกรมได้คืออะไร [1]

ตัวควบคุมแบบลำดับที่โปรแกรมได้หรือเรียกย่อ ๆ ว่า PC (Programmable Sequence Controller) เป็นอุปกรณ์ควบคุมการทำงานแบบลำดับ มีระบบการทำงานที่ควบคุมด้วยคอมพิวเตอร์ (Computer Control) มีไมโครโปรเซสเซอร์ (Microprocessor) เป็นหัวใจในการควบคุมการทำงาน PC จะทำงานตามคำสั่งที่โปรแกรมเข้าไปโดยผ่านทางตัวป้อนโปรแกรม (Programming Console) โปรแกรมคำสั่งที่ใช้มีอยู่หลายระบบด้วยกัน แต่ที่นิยมใช้กันมากก็คือ โปรแกรมแบบขั้นบันได (Ladder Diagram) ซึ่งเป็นแผนผังวงจรแบบดั่งเดิมที่ใช้กันมานานในระบบควบคุมการทำงานด้วยรีเลย์ ดังนั้นจึงทำให้ผู้ที่คุ้นเคยกับระบบเดิมสามารถใช้งานตัว PC ได้โดยไม่ยากนัก การป้อนโปรแกรมคำสั่งทำได้ง่ายโดยการแปลงวงจรแบบขั้นบันไดให้อยู่ในรูปของโปรแกรมที่ PC สามารถเข้าใจ PC จะอ่านสัญญาณจากอินพุท และจะให้สัญญาณตอบสนองไปยังเอาท์พุทตามการประมวลผลจากโปรแกรมควบคุมที่ป้อนให้กับมัน โปรแกรมควบคุมที่ป้อนให้กับมันจะเลียนแบบการทำงานของรีเลย์ ตัวตั้งเวลาและตัวนับ นอกเหนือไปจากนี้ PC ยังมีคำสั่งการทำงานเกี่ยวกับการจัดการกับข้อมูล (Data Handling) ซึ่งเป็นส่วนที่ทำให้ PC มีความสามารถเหนือกว่าวงจรรีเลย์ธรรมดามาก

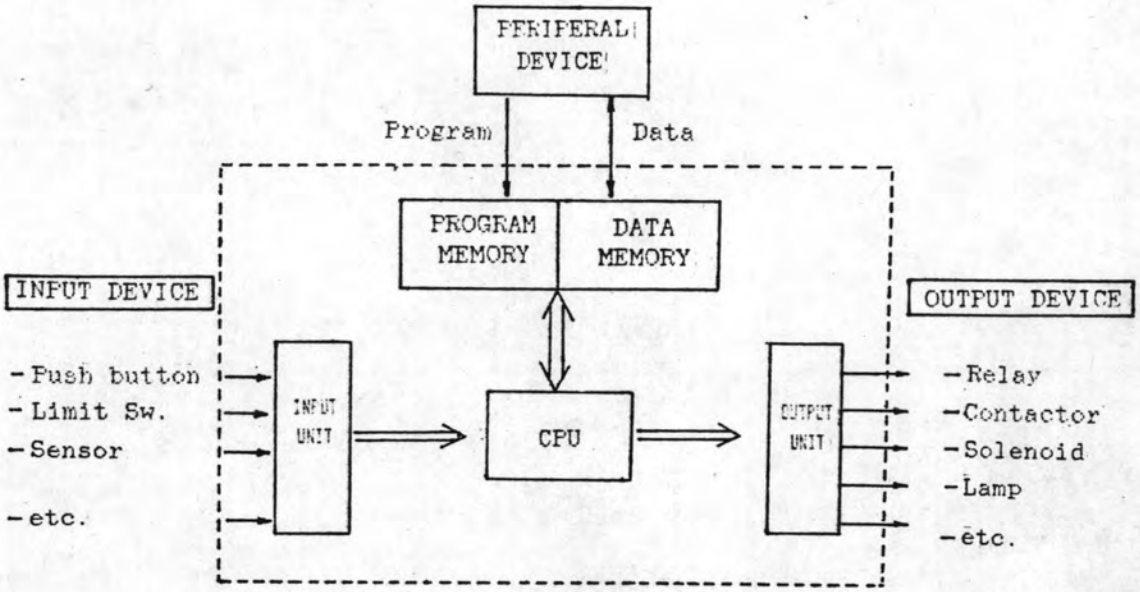
2.2 ประวัติความเป็นมาของ PC [1,5]

PC ถือกำเนิดมาจากบริษัท GENERAL MOTOR ซึ่งเป็นบริษัทผลิตรถยนต์รายใหญ่ที่สุดในอเมริกา บริษัทมักจะเผชิญกับปัญหาใหญ่ทุกครั้งเมื่อมีการออกแบบผลิตภัณฑ์ใหม่ ซึ่งจะต้องใช้ช่างเทคนิคทำการแก้ไขเปลี่ยนแปลงสายการผลิต ซึ่งเป็นระบบเดิมที่ควบคุมการทำงานด้วยรีเลย์ ทำให้สายการผลิตที่ผลิตอยู่ต้องหยุดการผลิตลง และต้องเสียเวลาในการเดินสายของวงจรควบคุมใหม่ เสียเวลาในการตรวจเช็คและทดสอบ ทำให้สิ้นเปลืองค่าใช้จ่ายเป็นจำนวนมากด้วยเหตุนี้ทางบริษัทจึงเสนอให้มีการพัฒนาตัว PC ขึ้น ในปลายปี 1960 PC ระบบแรกได้ถูกพัฒนาขึ้น

และนำมาใช้งานเป็นครั้งแรก และถูกพัฒนาให้สมบูรณ์แบบในราวปี 1969 บริษัทที่ทำการพัฒนาตัว PC นี้มีอยู่หลายบริษัทด้วยกัน เช่น MODICON ALLEN-BRADLEY TEXAS INSTRUMENTS และ SQUARE-D PC ในยุคแรกได้พัฒนาขึ้นโดยการนำมินิคอมพิวเตอร์ (MINICOMPUTER) เข้ามาใช้ควบคุมงานในโรงงาน การควบคุมในระยะแรกยังเป็นการควบคุมทางตรรก (Logic) แบบง่าย ๆ ตัวเครื่องมีขนาดใหญ่และราคาแพง ในปี 1970 ไมโครโปรเซสเซอร์เบอร์ 4004 ได้ถือกำเนิดขึ้นซึ่งเป็นจุดกำเนิดของยุคไมโครคอมพิวเตอร์ (Microcomputer) เทคโนโลยีของไมโครโปรเซสเซอร์ได้ถูกพัฒนาขึ้นตามลำดับ จนในปี 1974 ไมโครโปรเซสเซอร์เบอร์ 8080 ได้ถูกพัฒนาขึ้น ในปีต่อมาอุปกรณ์ทางอิเลคทรอนิคส์พวก LSI และ MEMORY CHIP ก็ได้รับการพัฒนาขึ้นมาเช่นกัน ในยุคนี้จึงเกิด PC ที่ควบคุมการทำงานแบบลำดับที่โปรแกรมได้ โดยใช้ไมโครโปรเซสเซอร์ (Microprocessor Based Programmable Sequence Controller) ทำให้ขนาดของ PC เล็กลง และมีความสามารถในการทำฟังก์ชั่นต่าง ๆ ได้มากขึ้นกว่าเดิม เช่น ฟังก์ชันทางคณิตศาสตร์ นอกจากนี้ยังมีความสามารถในการควบคุมระบบที่สลับซับซ้อนได้มากขึ้น เช่นมี ANALOG MODULES THERMOCOUPLE MODULES เป็นต้น ดังนั้นราคาของ PC จึงถูกลง แต่มีประสิทธิภาพการทำงานสูงขึ้น ปัจจุบันจึงมีผู้นิยมใช้กันมากและมีแนวโน้มว่าจะเพิ่มมากขึ้นเรื่อย ๆ

PC เริ่มเข้ามามีบทบาทในประเทศไทยเรา เมื่อประมาณ 4-5 ปีมาแล้ว โดยแฝงเข้ามาในรูปของตัวควบคุมเครื่องจักรที่สั่งเข้ามาใช้ในประเทศ ในปัจจุบันพบว่าเริ่มมีการใช้ PC เข้ามาควบคุมระบบในโรงงานอุตสาหกรรมมากขึ้น และประมาณกันว่าในอีกไม่กี่ปีข้างหน้า PC จะเข้ามาแทนที่ระบบเก่าเกือบทั้งหมด ด้วยเหตุนี้ตามสถาบันอุดมศึกษาหลายแห่งได้ทำการศึกษาและวิจัยตัว PC เป็นที่น่าเสียดายที่การศึกษาและวิจัยนั้น ยังไม่สามารถเจาะลึกลงไปในรายละเอียดการทำงานของ PC ได้อย่างจริงจัง การทดสอบได้ใช้แผ่นวงจรพิมพ์เดียว (Single Board) ซึ่งหลักการทำงานของแผ่นวงจรพิมพ์เดียวและ PC มีปรัชญาการออกแบบนำมาใช้งานที่แตกต่างกัน ดังนั้นการทดสอบและวิจัยที่ผ่าน ๆ มาจึงถูกจำกัดทำให้ขั้นโปรแกรมการควบคุม (Step) และจำนวนอินพุทและเอาท์พุทน้อยกว่าที่ควรจะเป็น เพราะจุดหลักในการทำโปรแกรมของแผ่นวงจรพิมพ์เดียวคือ การสแกนคีย์บอร์ด และตัวแสดงผล จึงทำให้ผลการวิจัยที่ผ่านมาไม่ประสบความสำเร็จเท่าที่ควร จึงเป็นจุดเริ่มต้นของการทำวิทยานิพนธ์นี้ โดยจะทำการศึกษาและเจาะลึกลงไปในรายละเอียดของปรัชญาการทำงานของ PC อย่างแท้จริง เพื่อทำให้ขั้นโปรแกรมการควบคุม (Step) และจำนวนอินพุทและเอาท์พุทมากสุด เต็มความสามารถของตัวไมโครปรเซสเซอร์ที่จะทำได้ นั่นคือการวิจัยจะทำการศึกษาทั้งโครงสร้างทางฮาร์ดแวร์ และโครงสร้างทางซอฟท์แวร์ของ PC เพื่อให้การทำงานของ PC เร็วที่สุดนั่นเอง

2.3 โครงสร้างของ PC [1]



รูปที่ 2.1 แสดงโครงสร้างของ PC

พิจารณารูปที่ 2.1 แสดงบล็อคไดอะแกรมของตัว PC การทำงานจะถูกควบคุมด้วยไมโครโปรเซสเซอร์ ดังนั้นการทำงานของระบบทั้งหมดจะถูกกำหนดด้วยโปรแกรมที่ป้อนให้กับมัน ระบบของ PC จะประกอบด้วยบล็อคต่าง ๆ ดังนี้คือ

2.3.1 ซีพียู (Central Processing Unit)

เป็นหน่วยประมวลผลกลางทำงานประสานกับทุกหน่วย โดยปกติทำหน้าที่ดังนี้

1. อ่านข้อมูลจากอินพุท แล้วนำข้อมูลมาประมวลผลตามโปรแกรมควบคุมที่เก็บไว้ในหน่วยความจำ ผลลัพธ์ที่ได้จากการประมวลผลจะถูกส่งออกไปที่เอาท์พุทเพื่อใช้ควบคุม อุปกรณ์ภายนอกอีกที

2. ทำหน้าที่รับส่งข้อมูลกับ PERIPHERAL DEVICES ซึ่งมีอยู่ด้วยกันหลายชนิด ดังจะกล่าวต่อไป

3. ตรวจเช็คโปรแกรมควบคุมที่ป้อนเข้าไปเก็บในหน่วยความจำว่าถูกต้องตาม แบบฟอร์มที่กำหนดไว้หรือไม่ ถ้าไม่ถูกต้องก็จะแจ้งให้ผู้ใช้ทราบเพื่อทำการแก้ไขให้ถูกต้อง

4. ตรวจเช็คแบตเตอรีที่ใช้เป็นแหล่งจ่ายไฟสำรองที่จ่ายให้กับหน่วยความจำที่เก็บโปรแกรมผู้ใช้ว่าระดับแรงดันปกติหรือไม่

2.3.2 หน่วยความจำข้อมูล (Data Memory)

เป็นหน่วยความจำที่ใช้เก็บข้อมูลที่เป็นสภาวะการทำงานของอุปกรณ์ภายใน เช่น สภาวะการทำงานของรีเลย์ สภาวะของอินพุท และ เอาท์พุท ว่าทำงานหรือไม่อย่างไร นอกจากนี้ยังใช้เก็บสภาวะการทำงานของตัวตั้งเวลา และตัวนับทุกตัวอีกด้วย

2.3.3 หน่วยความจำโปรแกรม (Program Memory)

เป็นหน่วยความจำที่ใช้เก็บโปรแกรมควบคุมการทำงานของผู้ใช้ โปรแกรมควบคุมนี้ผู้ใช้จะแปลงมาจากวงจรควบคุมแบบขั้นบันได (Ladder Diagram) แล้วป้อนเข้ามาเก็บในหน่วยความจำโดยผ่านตัวป้อนโปรแกรม (Programming Console) ซีพียู จะอ่านโปรแกรมควบคุมในหน่วยความจำทีละคำสั่งแล้วทำงานตามคำสั่งที่อ่านทันทีและจะทำคำสั่ง จนจบโปรแกรมควบคุม แล้วจะเริ่มต้นทำคำสั่งใหม่วนเวียนเช่นนี้ตลอดไป หน่วยความจำนี้ยิ่งมีมากเท่าไรก็สามารถบรรจุโปรแกรมควบคุมที่มีขนาดใหญ่และซับซ้อนได้มากยิ่งขึ้น

2.3.4 ภาคอินพุท (Input Module)

ทำหน้าที่รับสัญญาณจากสวิทช์และเซนเซอร์(Sensor) ต่าง ๆ ในระบบแล้วแปลงให้เป็นระดับสัญญาณไฟที่เหมาะสมที่ ซีพียูสามารถรับได้ สัญญาณไฟที่ป้อนเข้ามายังอินพุทอาจเป็นสัญญาณไฟ AC หรือ DC ก็ได้

ดังนั้นอินพุทอาจแบ่งออกได้เป็น 2 ชนิด คือ

1. **อินพุทชนิด AC (AC Input Type)** มักใช้เป็นแบบแยกสัญญาณ (Isolate) ออกจากซีพียู เหมาะสำหรับต่อกับหน้าสัมผัส (Contact Input) ที่มาจากโปรเซส (Process) เช่น Limit Switch และ Contactor เป็นต้น

2. **อินพุทชนิด DC (DC Input Type)** มีทั้งแบบกระแสเข้า(Current Sink input) และแบบกระแสออก (Current Source Input)

2.3.5 ภาคเอาท์พุท (Output Module)

เป็นตัวรับสัญญาณจากซีพียูแล้วทำหน้าที่ขยายสัญญาณออกให้มีขนาดใหญ่พอที่จะขับอุปกรณ์ภายนอก เช่น Contactor และ Solenoid Valve เป็นต้น นอกจากนี้ยังทำหน้าที่แยกสัญญาณภายในและสัญญาณภายนอกออกจากกัน เพื่อป้องกันระดับสัญญาณไฟสูงจากภายนอกเข้ามาทำความเสียหายแก่ซีพียู โดยปกติจะมีความสามารถในการขับโหลดด้วยกระแสประมาณ 2-5 Amp.

ภาคเอาท์พุทมีด้วยกันหลายชนิด การใช้งานจะต้องพิจารณาเลือกใช้ให้เหมาะสม คือ

1. เอาท์พุทชนิด AC (AC Output) มักจะเป็นแบบแยกสัญญาณ(Isolate)
   มีทั้งแบบหน้าสัมผัส (Contact Output)
   แบบไตรแอค (Triac Output)
   แบบไตรแอค (Triac Output) สามารถใช้งานได้ทนทานเหมาะสำหรับโหลดที่มีการเปิดปิดบ่อย ๆ

2. เอาท์พุทชนิด DC (DC Output) มีทั้งแบบแยกสัญญาณ (Isolate) และแบบไม่แยกสัญญาณ (Non-Isolate)
   มีทั้งแบบหน้าสัมผัส (Contact Output)
   แบบทรานซิสเตอร์ (Transistor Output)

2.3.6 PERIPHERAL DEVICES

เป็นอุปกรณ์แบบต่าง ๆ ที่ใช้อำนวยความสะดวกในการพัฒนาโปรแกรม มีอยู่ด้วยกันหลายชนิด และทำหน้าที่ต่างกัน เช่น

1. ตัวป้อนโปรแกรม (Programming Console) ทำหน้าที่ป้อนโปรแกรมควบคุมของผู้ใช้ที่นำไปใช้ในการควบคุมซีเควันซ์โดยนำไปเก็บไว้ในหน่วยความจำของ PC นอกจากนี้ยังใช้ในการแก้ไข (Edit) โปรแกรมควบคุมและใช้มอนิเตอร์ (Monitor) ดูสภาวะการทำงานภายในของรีเลย์ ตัวตั้งเวลา และตัวนับ

2. ซีอาร์ทีมอนิเตอร์ (CRT Monitor) ทำหน้าที่เช่นเดียวกับตัวป้อนโปรแกรม แต่ให้ความสะดวกในการโปรแกรมได้มากกว่า เนื่องจากสามารถมองเห็นรูปวงจรขั้นบันไดจากจอภาพได้ การป้อนและแก้ไขโปรแกรมทำได้รวดเร็วกว่า ตัวป้อนโปรแกรม แต่มีราคาแพงกว่ามาก

3. เทปคาสเซ็ท (Audio Cassette) ใช้ในการบันทึกโปรแกรมขั้นบัน

ได เพื่อเก็บรักษาไว้สำหรับการพัฒนาต่อไป เราสามารถโหลดโปรแกรมที่บันทึกเก็บไว้ในเทปคาสเซ็ท ถ่ายลงในหน่วยความจำของ PC ได้

4. **ตัวอัดโปรแกรม (EPROM Writer)** ใช้ในการอัดโปรแกรมขั้นบันไดไว้ในหน่วยความจำชนิด EPROM เมื่อนำ EPROM มาเสียบลงใน PC จะทำให้โปรแกรมขั้นบันไดอยู่ในหน่วยความจำอย่างถาวร

5. **เครื่องพิมพ์ (Printer)** ใช้ในการพิมพ์โปรแกรมเพื่อนำโปรแกรมมาตรวจสอบ และแก้ไข หรือเก็บเป็นเอกสาร

2.4 ระบบเลขฐาน (Numbering System) [5]

2.4.1 **ความสำคัญของระบบเลขฐาน** การใช้ระบบเลขฐานใน PC จะมีผลกระทบต่อความรู้สึกของผู้ใช้ คนโดยทั่วไปจะมีความรู้สึกเกี่ยวกับระบบเลขฐานที่แตกต่างกันออกไป คนที่มีความรู้และความเข้าใจในคณิตศาสตร์ และระบบเลขฐานเป็นอย่างดี ก็จะมีความรู้สึกสนุกสนานและเพลิดเพลินกับการใช้ PC ในการควบคุม ในทางตรงกันข้ามคนที่ไม่ชอบคณิตศาสตร์และระบบเลขฐานแล้ว จะมีความรู้สึกที่ไม่ค่อยดี และจะพยายามหลีกเลี่ยงการใช้เครื่องควบคุมที่ตัวเองคิดว่ายุ่งยาก ดังนั้นในการเลือกใช้ระบบเลขฐานใน PC จึงมีความสำคัญต่อความรู้สึกของผู้ใช้ ซึ่งจะต้องเลือกใช้ระบบเลขฐานให้เหมาะสม การเลือกใช้ระบบเลขฐานที่ไม่เหมาะสมจะทำให้ PC นั้น ๆ ไม่เป็นที่นิยม แม้ว่าจะมีฟังก์ชันการทำงานที่ดี สลับซับซ้อน และราคาถูกก็ตาม การทำความเข้าใจกับระบบเลขฐานจึงมีความสำคัญ เพราะในการควบคุมมักจะใช้เลขฐานเหล่านี้ในการรับและส่งข้อมูลระหว่างอินพุท เอาท์พุท หน่วยความจำ และ ซีพียู เพื่อใช้ในการประมวลผล

2.4.2 **ชนิดของระบบเลขฐาน** ระบบเลขฐานที่ใช้กันใน PCมีอยู่ด้วยกันหลายระบบ เช่น

1. **ระบบเลขฐานสิบ(Decimal System)** เป็นระบบเลขฐานที่ใช้กันอยู่ทั่วๆไปและเป็นที่รู้จักกันทั่วไป โดยระบบจะเริ่มต้นจากเลข "0" ซึ่งมีค่าน้อยสุดจนถึงเลข "9" ซึ่งมีค่ามากที่สุด

2. **ระบบเลขฐานสอง(Binary Numbering System)** เป็นระบบเลขฐานที่ใช้กันอยู่ทั่ว ๆ ไปในคอมพิวเตอร์ ค่าที่ใช้ในระบบมีเพียง 2 ค่าเท่านั้นคือ ค่า "0" และค่า "1" เนื่องจากเป็นระบบที่มีค่าเพียง 2 ค่า ดังนั้นค่าในคอลัมน์จึงแสดงตำแหน่ง หรือ แสดงน้ำหนัก (Weighting) ของการนับ

3. **ระบบเลข BCD (Binary-Coded Decimal System)** เป็นระบบเลขฐานที่รวมเอาระบบฐานสองและฐานสิบเข้าไว้ด้วยกัน โดยแต่ละหลักในระบบฐานสิบจะแทนด้วยระบบฐานสอง 4 หลัก

ระบบเลข BCD นี้เป็นระบบที่นิยมใช้กันมาก เพราะมันสามารถติดต่อกับคอมพิวเตอร์ด้วยเลขฐานสอง และในขณะเดียวกันรหัสโค๊ตของมันจะเป็นเลขฐานสิบสำหรับติดต่อกับผู้ใช้

4. **ระบบเลขฐานแปด(Octal Numbering System)** เป็นระบบเลขฐานที่ใช้ในคอมพิวเตอร์เช่นกัน ค่าที่ใช้นับมี 8 ค่า เริ่มจากค่า "0" จนถึงค่า "7" ระบบนี้มักนำมาใช้ใน PC โดยเฉพาะที่อินพุท และเอาท์พุทของ PC ที่ใช้ ซีพียู ขนาด 8 บิท เพราะสามารถอ่านค่า อินพุท และให้ค่าเอาท์พุทได้ทีละ 8 ค่า

5. **ระบบเลขฐานสิบหก(Hexadecimal Numbering System)** เป็นระบบเลขฐานอีกแบบหนึ่งที่นิยมใช้ในคอมพิวเตอร์ มีค่านับ 16 ค่า เริ่มจากค่า "0" ถึง "9" และค่านับที่ "10" ถึง "15" จะใช้อักษร A ถึง F แทน

2.5 ระบบเลขฐานที่ใช้ใน PC ของต่างประเทศ [1,2,3,4]

จากการศึกษา PC ของต่างประเทศ ทั้งขนาดเล็ก ขนาดกลาง และขนาดใหญ่ ระบบเลขฐานที่ใช้จะมีหลาย ๆ อย่างผสมกันขึ้นอยู่กับขนาดของ PC ใน PC ขนาดเล็กที่ใช้คำสั่งไม่ซับซ้อน มักจะใช้ระบบฐานแปด และ BCD ผสมกัน ส่วนขนาดใหญ่มักมีระบบฐานสองและฐานสิบหกเข้ามาเกี่ยวข้องด้วย ดังนั้นในระบบขนาดใหญ่ผู้ใช้จึงมักมีความรู้เกี่ยวกับระบบเลขฐานต่าง ๆ เป็นอย่างดีแล้ว ตารางที่ 2.1แสดงการเปรียบเทียบการใช้ระบบเลขฐานในส่วนต่างๆ ของ PC จากต่างประเทศที่มีจำหน่ายในท้องตลาด

ตารางที่ 2.1 แสดงการเปรียบเทียบการใช้ระบบเลขฐานใน PC ของต่างประเทศ

ผู้ผลิต	ขนาด	รุ่น	ระบบเลข	หมายเหตุ
FUJI ELECTRIC	เล็ก	FUJILOG-UT	BCD, ฐานแปด	-ใช้ทั้ง2ระบบในการเขียนคำสั่ง -ใช้ระบบ BCD ในการมอนิเตอร์ดูค่าเวลาของตัวตั้งเวลาและค่านับของตัวนับ
	ใหญ่	MICREX-F	BCD, ฐานสอง และฐานสิบหก	-ใช้ระบบ BCD และฐานสิบหกในการเขียนคำสั่ง -การมอนิเตอร์ดูค่าเวลาของตัวตั้งเวลาและค่านับของตัวนับดูได้ทั้งระบบ BCD และฐานสอง
OMRON	เล็ก	SYSMAC-C20P SYSMAC-C28P SYSMAC-C40P	BCD	-ใช้ BCD ทั้งระบบ
	กลาง	SYSMAC-C120 SYSMAC-C250 SYSMAC-C500	BCD	-ใช้ BCD ทั้งระบบ
FANUC & GE	เล็ก	SERIES 90-10	BCD, ฐานสอง และฐานแปด	-ใช้ BCD และฐานแปดในการเขียนคำสั่ง -ใช้ฐานสองในการมอนิเตอร์ดูค่าเวลาของตัวตั้งเวลาและค่านับของตัวนับ
MITSUBISHI ELECTRIC	เล็ก	MELSEC F-20M MELSEC F-40M	BCD, ฐานแปด	-เหมือน FUJI ขนาดเล็ก

จากตารางการเปรียบเทียบจะเห็นว่าระบบเลขฐานที่นิยมใช้กันมาก คือระบบเลข BCD และระบบเลขฐานแปด ดังนั้นในการทำวิทยานิพนธ์ จึงเลือกใช้ระบบเลขฐานเป็น BCD และระบบเลขฐานแปด ซึ่งเป็นระบบเลขฐานที่เหมาะสำหรับ PC ขนาดเล็ก ซึ่งตรงกับวัตถุประสงค์ในการทำวิทยานิพนธ์นี้ จุดที่จะนำระบบเลขฐานไปใช้ได้แก่ การป้อนโปรแกรมผู้ใช้ จากรูปที่ 2.2 เป็นตัวอย่างโปรแกรมที่ใช้ป้อนใน PC

```
MNEMONIC    OPERAND          COMMENT
  LD          107     ; LOAD STATUS RELAY CHANNEL 10 POSITION 7
  AND         115     ; AND STATUS RELAY CHANNEL 11 POSITION 5
  OUT         200     ; OUT RESULT TO RELAY CHANNEL 20 POSITION 0
```

รูปที่ 2.2 แสดงตัวอย่างโปรแกรมควบคุม

จากตัวอย่าง OPERAND จะประกอบด้วยระบบเลขฐาน BCD และฐานแปด โดย OPERAND จะถูกแบ่งออกเป็น 2 ส่วนคือแชนแนล (Channel) ซึ่งเริ่มจาก 00~99 ในระบบเลขฐาน BCD และอีกส่วนหนึ่งเริ่มจาก 0~7 คือตำแหน่ง (Position) ที่จะใช้ระบบเลขฐานแปด การใช้ระบบผสมแบบนี้จะทำให้ระบบมีโครงสร้างของข้อมูลที่ง่าย และจะให้ความรู้สึกง่ายต่อการใช้งานของผู้ใช้โดยทั่วไป เพราะระบบเลขที่ใช้คุ้นเคยดีอยู่แล้ว ส่วนการมอนิเตอร์ดูค่าเวลาของตัวตั้งเวลา และค่านับของตัวนับจะใช้ระบบเลขเป็น BCD

2.6 ระบบการโปรแกรม (Programming System) [7]

PC ทำงานได้ด้วยโปรแกรม และภาษาที่สามารถป้อนให้กับ PC มีด้วยกันหลายภาษา เช่น

2.6.1 BOOLEAN LOGIC EQUATION

รูปที่ 2.3 แสดงตัวอย่างของสมการบูลีน ใช้เขียนขึ้นเพื่อแสดงการคำนวณค่าของอินพุท 4 ตัวจากสมการ แล้วส่งผลลัพธ์ไปยังเอาท์พุท ภาษานี้มีประโยชน์เพื่อนำมาใช้กระทำทางตรรกแบบง่าย ๆ ได้มีการพัฒนาสมการบูลีนให้เป็นภาษาบูลีน(Boolean Language) โดยให้มีความสามารถในการกระทำฟังก์ชันที่สลับซับซ้อน เพื่อนำมาใช้กับ PC เช่นมีการเพิ่มให้มีตัวตั้งเวลา (TIMER) และตัวนับ (COUNTER) เป็นต้น

2.6.2 THE MNEMONIC LANGUAGE

รูปที่ 2.4 แสดงตัวอย่างของภาษา MNEMONIC ที่นำไปใช้เขียนเป็นโปรแกรมควบคุม โดยจะเป็นโค๊ดคำสั่งจริงที่ใช้กับตัวควบคุม ภาษานี้สามารถเขียนฟังก์ชันเพิ่มเติมเพื่อนำไปควบคุมได้อีก แต่จะต้องมีความรู้เกี่ยวกับการทำงานภายในเครื่องเป็นอย่างดี และการอ่านผลลัพธ์ของโปรแกรมทำได้ค่อนข้างยาก

2.6.3 THE LOGIC DIAGRAM LANGUAGE

ภาษาที่เขียนเป็นวงจรโดยใช้เกทต่าง ๆ นำมาต่อกันเป็นวงจรควบคุมที่ต้องการ ดังแสดงในรูปที่ 2.5 ภาษานี้เป็นภาษาที่ง่ายในการแปลงให้เป็นโค๊ดที่ PC เข้าใจ ถ้าวงจรควบคุมที่ต่อเป็นแบบง่ายๆแต่ความสามารถในการขยายให้ทำฟังก์ชันที่ซับซ้อนทำได้ค่อนข้างยาก

2.6.4 LADDER DIAGRAM LANGUAGE

รูปที่ 2.6 แสดงตัวอย่างของ LADDER DIAGRAM ซึ่งเหมือนกับวงจรที่ใช้ควบคุมระบบรีเลย์เป็นภาษาที่นิยมใช้ในการโปรแกรมตัวควบคุมอย่างมากเพราะ

1. การแปลงภาษาจากวงจรที่ใช้ควบคุมระบบรีเลย์ ให้มาเป็นภาษาเพื่อป้อนให้ PC ทำได้ง่าย
2. การตรวจซ่อม เพื่อหาจุดผิดพลาดของวงจรทำได้ง่าย
3. เป็นภาษาที่ง่ายต่อการอ่าน และทำความเข้าใจ
4. สามารถขยายให้ทำฟังก์ชันที่สลับซับซ้อนได้ง่าย โดยการเพิ่ม FUNCTION BLOCK ลงไปใน LADDER DIAGRAM ดังแสดงในรูปที่ 2.7

2.6.5 LADDER DIAGRAM WITH DATA FLOW

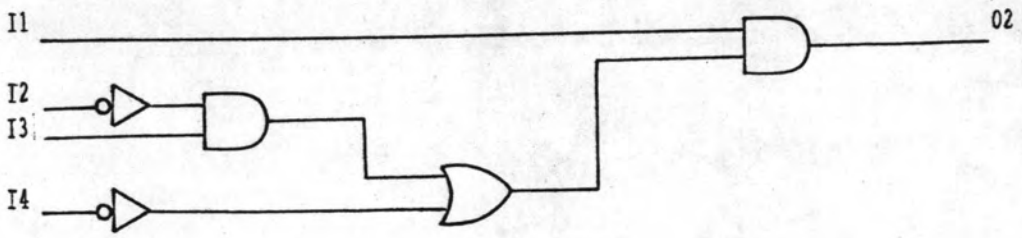
แม้ว่า LADDER DIAGRAM สามารถเพิ่ม FUNCTION BLOCK เพื่อให้สามารถทำฟังก์ชันที่ซับซ้อนได้ แต่ค่าต่าง ๆ ที่แสดงภายในบล็อค สามารถแสดงข้อมูลต่าง ๆ ได้เพียงค่าเดียว ในบางโปรเซส(Process) มีการนำค่าของข้อมูลมาคำนวณหาผลลัพธ์ โดยที่ค่าของข้อมูลมีการเปลี่ยนแปลงอยู่ตลอดเวลา ดังนั้นการเขียนโดย FUNCTION BLOCK จึงไม่สามารถแสดงการเปลี่ยนแปลงของข้อมูลได้ แต่เราสามารถเขียนแทนได้ด้วย LADDER DIAGRAM WITH DATA FLOW ดังแสดงในรูปที่ 2.8 ภาษานี้เหมาะสำหรับนำไปใช้ในกระบวนการควบคุมที่ยุ่งยากมาก ๆ

```
I1 & (/I2  &  I3  +  /I4)  =  O2
```

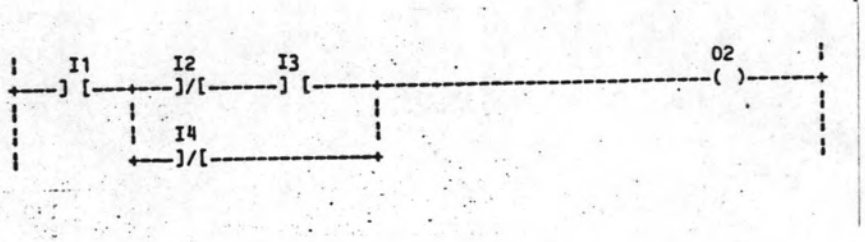
รูปที่ 2.3 แสดงตัวอย่างของสมการบูลีน

```
XIC     I1      eXamine Input Closed I1
BST             Branch STart
XIO     I2      eXamine Input Open I2
XIC     I3      eXamine Input Closed I3
BST             Branch STart
XIO     I4      eXamine Input Open I4
BND             Branch eND
OTE     O2      OuTput Energize O2
```

รูปที่ 2.4 แสดงตัวอย่างของภาษา MNEMONIC



รูปที่ 2.5 แสดงตัวอย่างภาษา LOGIC DIAGRAM



รูปที่ 2.6 แสดงตัวอย่างของภาษา LADDER DIAGRAM

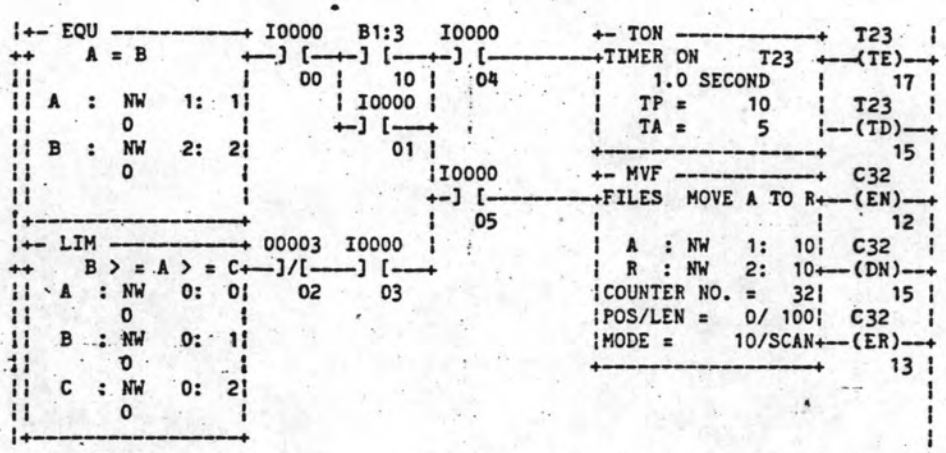


FIGURE 4: Example rung of a program in ladder diagram form with functional blocks.

รูปที่ 2.7 แสดงตัวอย่างของภาษา LADDER DIAGRAM WITH FUNCTIONAL BLOCKS

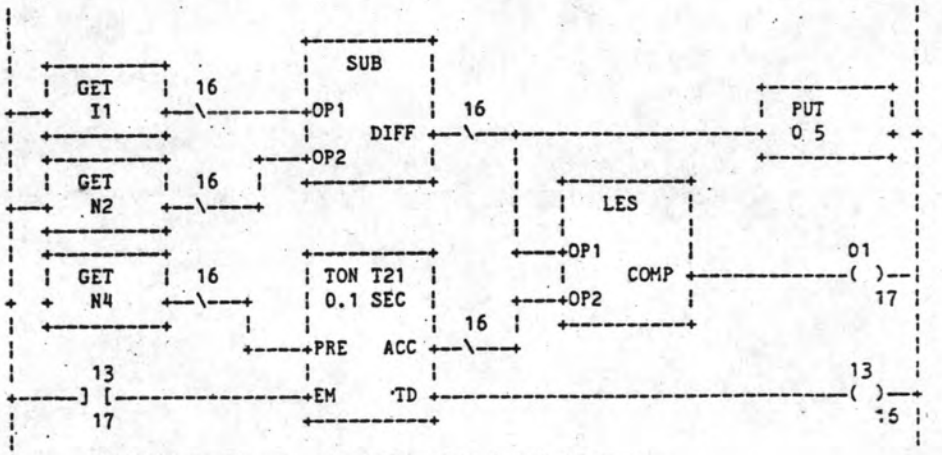


FIGURE 5: Example of a data flow (bus) diagram

รูปที่ 2.8 แสดงตัวอย่างของภาษา LADDER DIAGRAM WITH DATA FLOW

016139

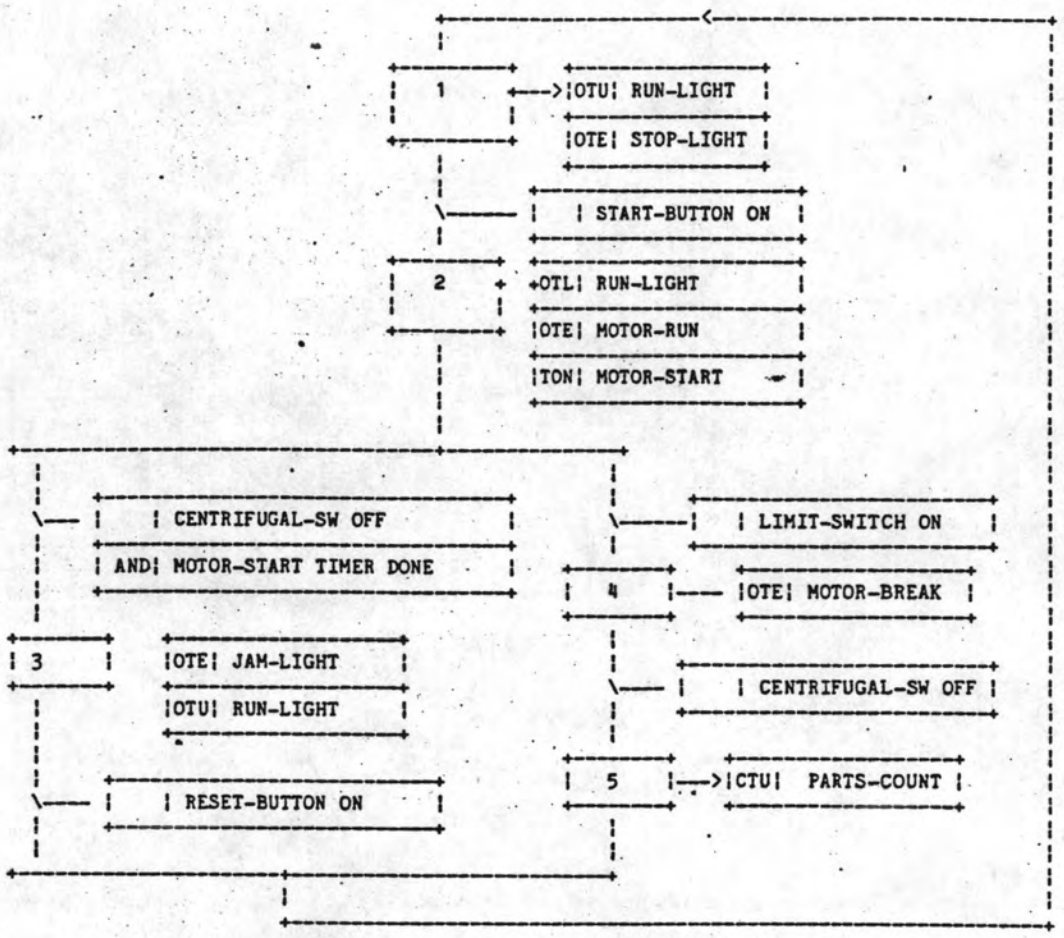


FIGURE 6: Example of a function chart type of language

รูปที่ 2.9 แสดงตัวอย่างของภาษา FUNCTION CHART

2.6.6 FUNCTION CHART

รูปที่ 2.9 แสดงตัวอย่างของ FUNCTION CHART การทำงานจะเป็นแบบลำดับขั้น โดยจะทำทีละขั้นตามกระบวนการควบคุมใน CHART ในแต่ละขั้นของ FUNCTION CHART จะมี INPUT BLOCK เพื่อตรวจสอบเงื่อนไข ถ้าเงื่อนไขที่ตรวจสอบเป็นไปตามเงื่อนไขที่ได้วางไว้ จะมีการกระทำในขั้นตอนต่อไป ส่วนเอาท์พุทในแต่ละขั้นตอนจะควบคุมให้มีการทำงานตามเงื่อนไขที่กำหนดไว้ และจะมีการแยกไปในกิ่งใดกิ่งหนึ่งขึ้นอยู่กับเงื่อนไขที่พบ

จากการเปรียบเทียบภาษาต่าง ๆ จากตารางที่ 2.2 จะเห็นว่าภาษา LADDER DIAGRAM WITH FUNCTIONAL BLOCK เป็นภาษาที่ดีที่สุด เพราะสามารถนำมาใช้เป็นภาษาของ PC ดังได้กล่าวมาแล้ว ดังนั้นในวิทยานิพนธ์นี้ ภาษาที่ใช้ใน PC จะเป็น LADDER DIAGRAM LANGUAGE ส่วนการแปล LADDER DIAGRAM LANGUAGE ให้เป็นโปรแกรมเพื่อป้อนให้แก่ PC จะกล่าวรายละเอียดในบทต่อ ๆ ไป

ตารางที่ 2.2 แสดงการเปรียบเทียบข้อดีและข้อเสียของแต่ละภาษา

ภาษา	ข้อได้เปรียบ	ข้อเสียเปรียบ
BOOLEAN LOGIC LANGUAGES	เหมาะสำหรับวิศวกรใช้	ขยายเพื่อทำฟังก์ชันอื่นได้ยาก
MNEMONIC LANGUAGES	ง่ายต่อการขยายฟังก์ชัน	จะต้องรู้การทำงานภายในเครื่องควบคุมอย่างดี
LOGIC SCHEMATIC LANGUAGES	ใช้สัญลักษณ์ทางคณิตศาสตร์มาต่อเพื่อใช้ในการควบคุม	ขยายเพื่อทำฟังก์ชันอื่นได้ยาก
LADDER DIAGRAM WITH FUNCTIONAL BLOCK	แสดงรายละเอียดของวงจรได้ดี ง่ายต่อการอ่านวงจร ง่ายต่อการขยายฟังก์ชัน	โดยหลักการแล้วไม่ได้ออกแบบมาเพื่อการควบคุมแบบลำดับ
DATA FLOW DIAGRAM	เหมาะสำหรับขบวนการกระทำที่เกี่ยวข้องกับข้อมูลมากๆ	ยังพัฒนาไม่สมบูรณ์แบบ
FUNCTION CHART	ดีสำหรับการควบคุมที่เป็นแบบลำดับ	ยังพัฒนาไม่สมบูรณ์แบบ