บทที่ 2

ฟังก์ชั่นชาร์ต

2.1 ความเป็นมา

ตั้งแต่สมัยที่รีเลย์ถูกประดิษฐ์ขึ้นมา การควบคุมการทำงานแบบลำดับในอุตสาหกรรม ไม่ว่าจะเป็นการควบคุมการทำงานของเครื่องจักร การควบคุมการประกอบ ตลอดจนการควบคุมการผลิตของโรงงาน ต่างก็ใช้วงจรรีเลย์เป็นตัวควบคุมแทบทั้งสิ้นการออกแบบวงจรรีเลย์มักจะเขียนเป็นแผนภาพ รูปแบบหนึ่งที่เป็นที่นิยมได้แก่แผนภาพแลดเดอร์ ซึ่งใช้แสดงเงื่อนไขการทำงานของวงจรรีเลย์แต่ละตัว

ต่อมาเมื่อเทคโนโลยีคอมพิวเตอร์ก้าวหน้าขึ้น จึงเกิดการพัฒนาเครื่องควบคุมชนิดโปรแกรมได้ เพื่อนำมาใช้แทนวงจรรีเลย์ ทำให้เกิดการเปลี่ยนแปลงของเทคโนโลยีจากระบบรีเลย์ ไปสู่ไมโครคอมพิวเตอร์สำหรับงานอุตสาหกรรม แม้ฮาร์ดแวร์จะเปลี่ยนไป แต่แผนภาพแลดเดอร์ก็ยังเป็นที่นิยมใช้ในการออกแบบอยู่ ในคราวนี้จึงสามารถนำแผนภาพแลดเดอร์ของวงจรรีเลย์ที่ออกแบบไว้ไปป้อนเข้าเครื่องควบคุมชนิดโปรแกรมได้ ได้โดยตรง ทำให้การแก้ไขวงจร การเปลี่ยนค่าต่างๆ ทำได้โดยง่าย

เหตุที่แผนภาพแลดเดอร์เป็นที่นิยมใช้ก็เนื่องจากข้อดี เช่น สามารถแสดงเงื่อนไขการทำงานของเครื่องจักรได้ง่าย วิศวกรและช่างเทคนิคที่คุ้นเคยกับวงจรรีเลย์ สามารถทำความเข้าใจได้ทันที ขณะเดียวกันแผนภาพแลดเดอร์ก็มีจุดอ่อนอยู่หลายประการ ได้แก่

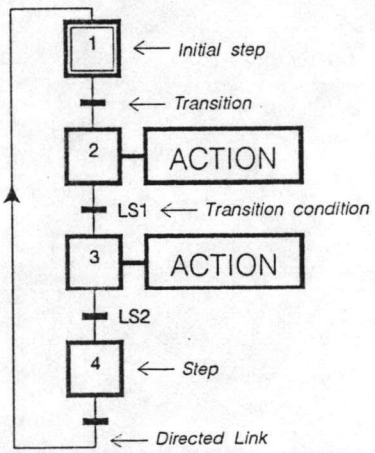
1. อ่านได้ยาก เข้าใจได้ยาก ยิ่งมีขนาดใหญ่ ยิ่งดูซับซ้อน
2. การออกแบบวงจร ต้องใช้เวลาและประสบการณ์ค่อนข้างมาก
3. ไม่สามารถแสดงลำดับการทำงานของอุปกรณ์แต่ละตัวได้อย่างชัดเจน
4. ในกรณีกลุ่มที่ทำงานร่วมกัน ประกอบด้วยบุคคลจากหลายๆ ฝ่ายก็ไม่สามารถใช้แผนภาพแลดเดอร์สื่อสารกันได้เข้าใจทุกคน
5. การแก้ไขและการตรวจหาข้อผิดพลาดทำได้ยาก

จึงได้มีการพัฒนารูปแบบการเขียนโปรแกรมใหม่ขึ้นมาโดยชาวฝรั่งเศส เพื่อทดแทนจุดอ่อนของแผนภาพแลดเดอร์ข้างต้น และเรียกว่า "กราฟเซ็ท (GRAFCET)" กราฟเซ็ทมีลักษณะการเขียนโปรแกรมคล้ายโฟล์วชาร์ต (Flow Chart) และสามารถใช้เขียนแสดงลำดับการควบคุมได้ดี ประเทศฝรั่งเศสจึงตั้งกราฟเซ็ทเป็นมาตรฐานสำหรับใช้เขียนโปรแกรมของ

ระบบควบคุมหมายเลข NF C 03-190 บริษัท เทเลแมคคานิค (TELEMECANIQUE) ผู้ผลิตเครื่องควบคุมชนิดโปรแกรมได้ของประเทศฝรั่งเศส จึงได้พัฒนาเครื่องของตนให้ใช้กราฟเซ็ทเขียนโปรแกรมได้ ประเทศเยอรมันเองก็มีการพัฒนารูปแบบของภาษาที่คล้ายโฟล์ชาร์ตออกมาและตั้งเป็นมาตรฐานหมายเลข DIN 40719 ต่อมา IEC (International Electrotechnical Commission) เล็งเห็นว่าน่าจะมีมาตรฐานสำหรับการเขียนโปรแกรมในลักษณะนี้สำหรับยุโรป จึงได้ร่างข้อเสนอโดยอ้างอิงตามกราฟเซ็ท ส่งไปยังประเทศต่างๆ บริษัท ซีเมนส์ (SIEMENS) ผู้ผลิตเครื่องควบคุมชนิดโปรแกรมได้ของประเทศเยอรมัน จึงได้พัฒนาภาษาลักษณะเช่นนี้ขึ้นมาโดยอ้างอิงตามข้อเสนอของ IEC และมาตรฐาน DIN 40719 และเรียกภาษาใหม่นี้ว่า"กราฟไฟว์ (GRAPH5)" เครื่องควบคุมชนิดโปรแกรมได้รุ่นใหม่ๆ ของซีเมนส์ ก็จะสามารถเขียนโปรแกรมด้วยภาษากราฟไฟว์ได้ ในปี 1988 IEC ก็สามารถตั้งมาตรฐานสำหรับภาษารูปแบบนี้ได้สำเร็จ โดยใช้รหัส "IEC848" และเรียกว่า "ฟังก์ชั่นชาร์ต (Function Chart)

2.2 ทฤษฎี

ในการควบคุมเชิงวิศวกรรม สามารถแบ่งการควบคุมได้เป็น 2 รูปแบบ[1] ได้แก่ การควบคุมเชิงตรรก (Logic Control) และการควบคุมลำดับการทำงาน (Sequence Control) ข้อแตกต่างก็คือ การควบคุมเชิงตรรกจะอธิบายความสัมพันธ์ระหว่างสัญญาณเข้าและสัญญาณออกของตัวควบคุม ในขณะที่การควบคุมลำดับการทำงาน จะเน้นไปที่ฟังก์ชั่นการควบคุม ซึ่งจังหวะเวลาของสัญญาณเข้าและสัญญาณออกเป็นเรื่องสำคัญ ซึ่งพอจะให้คำจำกัดความของการควบคุมลำดับการทำงานได้ว่า เป็นรูปแบบหนึ่งของการควบคุม ซึ่งจะบังคับให้เกิดการทำงานอย่างเป็นลำดับเป็นขั้นเป็นตอน และจากขั้นตอน (Step) หนึ่ง จะผ่านไปยังอีกขั้นตอนหนึ่งได้ จะขึ้นอยู่กับเงื่อนไขการเปลี่ยนขั้นตอน (Step Enabling Condition) ระหว่างขั้นตอนทั้งสอง



รูปที่ 2.1 : องค์ประกอบเบื้องต้น

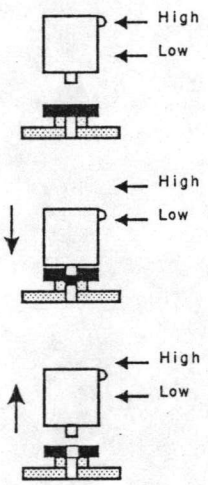
1. SIEMENS. SIEMATIC S5 GRAPH5 Introduction. pp. 1-2.

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ฟังก์ชั่นชาร์ตเป็นภาษาที่ถูกออกแบบมาเพื่อใช้เขียนโปรแกรมสำหรับควบคุมลำดับการทำงาน โดยเป็นภาษารูปภาพ (Graphical Language) จึงสามารถใช้เขียนแสดงลำดับการทำงานได้โดยง่าย ฟังก์ชั่นการควบคุมจะถูกแบ่งออกเป็นขั้นตอนการทำงานต่างๆ โดยจะมีการทำงานที่ขึ้นกับเงื่อนไขการเปลี่ยนขั้นตอน ขั้นตอนการทำงานต่างๆ จะเรียกว่า "สเต็ป (Step)" ซึ่งเขียนแทนด้วยสัญลักษณ์ รูปสี่เหลี่ยม(ตามมาตรฐาน IEC 848 แนะนำว่าควรใช้รูปสี่เหลี่ยมจัตุรัส) และมีหมายเลขของสเต็ปเขียนแสดงไว้ตรงกลาง การทำงานของสเต็ปนั้น (Action) จะเขียนแสดงไว้ในสี่เหลี่ยมข้างๆ สเต็ปที่จะทำงานเป็นสเต็ปแรก เรียกว่า "สเต็ปเริ่มต้น" (Initial Step) ให้เขียนรูปสี่เหลี่ยมจัตุรัสด้วยเส้นคู่ เงื่อนไขการเปลี่ยนขั้นตอนจะเขียนไว้ใน "ทรานสิชั่น" (Transition) ซึ่งมีสัญลักษณ์เป็นเส้นตรงที่มีขีดขวางกลาง แต่ละสเต็ปจะเชื่อมต่อกันผ่านทรานสิชั่น และการเชื่อมต่อระยะไกลๆ จะใช้เส้นเชื่อมต่อ (Directed link) ช่วย

เพื่อให้เข้าใจหลักการของฟังก์ชั่นชาร์ตได้ชัดเจนยิ่งขึ้น จะขอยกตัวอย่าง ซึ่งจะเปรียบเทียบวิธีแผนภาพแลดเดอร์ กับวิธีฟังก์ชั่นชาร์ต

ตัวอย่างที่ 1 : การควบคุมเครื่องปั๊มโลหะ[2]



รูปที่ 2.2 : การทำงานของเครื่องปั๊ม

ขั้นตอนที่ 1 : เครื่องหยุดที่ตำแหน่ง "High" จนกว่าจะได้รับสัญญาณ "Start";

ขั้นตอนที่ 2 : เครื่องปั๊มเคลื่อนลงจนกว่าจะเคลื่อนมาถึงตำแหน่ง "Low"

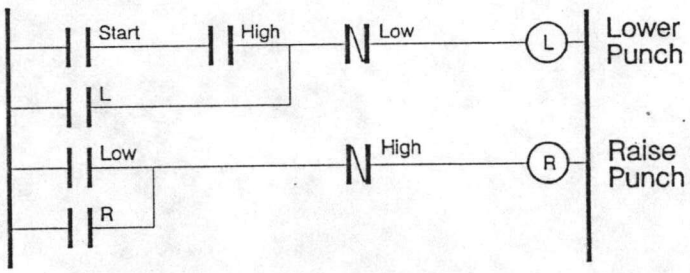
ขั้นตอนที่ 3 : เครื่องเคลื่อนขึ้น จนกว่าจะถึงตำแหน่ง "High"

ในขณะเริ่มต้นเครื่องปั๊มจะอยู่ในตำแหน่ง "High" ซึ่งตรวจสอบได้ด้วยลิมิตสวิตช์ตัวบน และเมื่อปุ่มสตาร์ทถูกกด เครื่องจะเคลื่อนลงมาข้างล่าง จนกระทั่งถึงตำแหน่ง "Low" ก็จะเคลื่อนกลับขึ้นข้างบนจนถึงตำแหน่ง "High" ก็จะหยุด

---

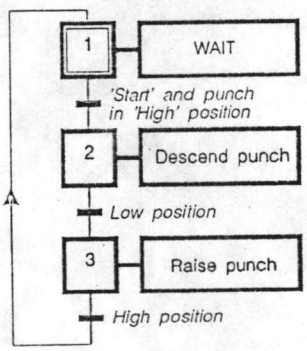
2. กฤษดา วิศวธีรานนท์. ฟังก์ชั่นชาร์ต ภาษาใหม่สำหรับการโปรแกรม PLC. วารสารเทคนิค ฉบับที่ 86, 2536. หน้า 81-88.

ครบรอบการทำงาน การเขียนโปรแกรมโดยใช้แผนภาพแลดเดอร์ จะมองที่เอาต์พุตเป็นหลัก โดยตรวจดูเงื่อนไขต่างๆ ที่มีผลต่อเอาต์พุตนั้น เช่นในกรณีนี้จะมี 2 เอาต์พุต ได้แก่ การสั่งให้เครื่องปั๊มเคลื่อนลง และการสั่งให้เครื่องปั๊มเคลื่อนขึ้น เมื่อมองที่การสั่งให้เคลื่อนลงก็จะมีเงื่อนไขต่างๆที่มีผลได้แก่ เครื่องอยู่ที่ตำแหน่งบนและปุ่มสตาร์ทถูกกดและต้องคำนึงถึงเมื่อปุ่มสตาร์ทถูกปล่อยด้วย เพราะสัญญาณจากปุ่มสตาร์ทจะหมดลง จึงต้องนำเอาต์พุตมาคงค่า (Hold) ตัวเองไว้ นอกจากนั้นจะต้องใส่เงื่อนไขที่จะหยุดการทำงานของเอาต์พุตรวมไว้ด้วย ซึ่งได้แก่สัญญาณจากลิมิตสวิตช์ "Low" แผนภาพแลดเดอร์ที่เสร็จแล้วแสดงดังรูป 2.3



รูปที่ 2.3 : แผนภาพแลดเดอร์

ด้วยวิธีฟังก์ชั่นชาร์ต[3] จะมีสามสเต็ปที่ชัดเจนดังแผนภาพในรูปที่ 2.4 กล่าวคือ เมื่อเริ่มการทำงานจะเข้าไปอยู่ที่สเต็ปเริ่มต้น (Initial Step) และรอคอยสัญญาณสตาร์ทเมื่อเงื่อนไขในทรานสิชั่นเป็นจริง จะเปลี่ยนสเต็ปจากสเต็ป 1 ไปสเต็ป 2 ทำให้เครื่องเคลื่อนลงมา เมื่อเครื่องเคลื่อนถึงตำแหน่ง "Low" ซึ่งเป็นเงื่อนไขสำหรับเปลี่ยนไปสเต็ป 3 ก็จะมีการเปลี่ยนสเต็ปไป เครื่องจะเคลื่อนขึ้น จนถึงตำแหน่ง "High" ก็จะเป็นเงื่อนไขให้เปลี่ยนไปที่สเต็ป 1 ใหม่

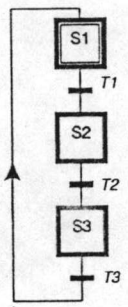


รูปที่ 2.4 : ฟังก์ชั่นชาร์ต

---

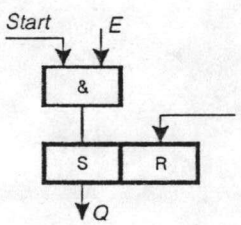
3. Michale Lloyd. GRAFCET : Graphical Function Chart Programming. TELEMECANIQUE. pp. 51-56.

จะเห็นได้ว่าการเขียนโปรแกรมด้วยวิธีฟังก์ชั่นชาร์ต สามารถเขียนได้ง่ายกว่าวิธีแผนภาพแลดเดอร์มาก และยังสามารถแสดงลำดับการทำงานได้ชัดเจนอีกด้วย ข้อดีของภาษาฟังก์ชั่นชาร์ตอีกประการหนึ่งก็คือ สามารถถูกแปลไปเป็นสมการบูลลีน (Boolean Equation) ได้ จึงทำให้ภาษานี้ไม่ยึดติดกับฮาร์ดแวร์ใด กล่าวคือ เมื่อจะนำฟังก์ชั่นชาร์ตไปใช้กับฮาร์ดแวร์ใด ก็ให้แปลสมการบูลลีนที่ได้ไปเป็นภาษาของฮาร์ดแวร์นั้น เช่นในงานวิจัยนี้หลังจากแปลภาษาฟังก์ชั่นชาร์ตไปเป็นสมการบูลลีนแล้ว ก็จะแปลสมการบูลลีนเหล่านั้น ไปเป็นภาษานีมอนิค (Mnemonic Language) ของเครื่องควบคุมชนิดโปรแกรมได้ ซึ่งภาษานีมอนิค ก็คือการแสดงการเชื่อมต่อของฮาร์ดแวร์นั่นเอง



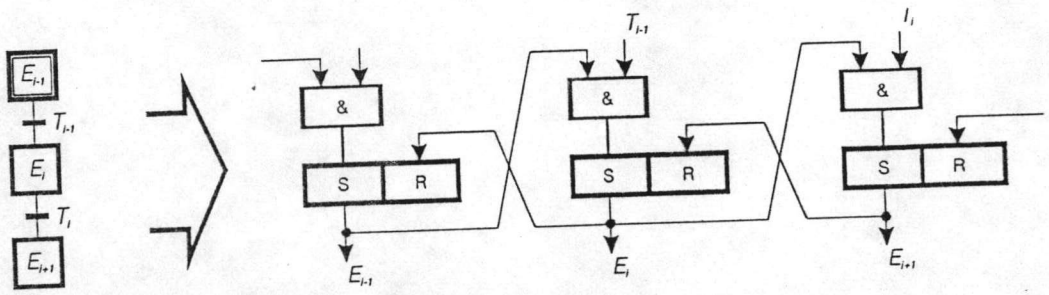
รูปที่ 2.5 : ตัวอย่างลำดับ

การแปลภาษาฟังก์ชั่นชาร์ตไปเป็นสมการบูลลีน สามารถอธิบายให้เข้าใจได้โดยง่าย หากเข้าใจคุณสมบัติ และกลไกการทำงานของการควบคุมแบบลำดับก่อน จากแผนภาพในรูปที่ 2.5 การทำงานจะเริ่มต้นที่สเต็ป S1 และเมื่อเงื่อนไขการเปลี่ยนสเต็ป T1 เป็นจริง (มีสัญญาณอินพุตที่ตรงตามเงื่อนไขT1 เกิดขึ้น) สเต็ป S2 ก็จะทำงาน และสเต็ป S1 จะหยุดการทำงานทันที สเต็ป S2 จะทำงานไปเรื่อยๆ จนกว่า เงื่อนไขการเปลี่ยนสเต็ป T2 จะเป็นจริง สเต็ป S3 ก็จะทำงานไปจนกว่าเงื่อนไข T3 จะเป็นจริง การทำงานก็จะวนกลับไปเริ่มต้นที่สเต็ป S1 ใหม่



รูปที่ 2.6 แบบจำลองของสเต็ป

ด้วยความเข้าใจในกลไกการทำงาน จะเห็นว่าสเต็ปประพฤติตัวในลักษณะคล้ายฟลิปฟลอป[4] (Flip-flop) จึงสามารถสร้างแบบจำลองของสเต็ปได้ดังรูปที่ 2.6 กล่าวคือสเต็ปจะทำงาน (Set) เมื่อเงื่อนไขอินพุตเป็นจริง ซึ่งได้แก่ สัญญาณ start ซึ่งเป็นสัญญาณกระตุ้นจากสเต็ปก่อนหน้า และสัญญาณ E ซึ่งเป็นสัญญาณที่ใช้เป็นเงื่อนไขการเปลี่ยนสเต็ปเป็นจริงพร้อมกันทั้งสองสัญญาณ สเต็ปก็จะแอ็กทีฟ (active) ให้เอาต์พุต Q ซึ่งเป็นการทำงานของสเต็ปนั่นเอง สเต็ปจะยังคงทำงานไปเรื่อยๆ (Latch) จนกว่าจะได้รับสัญญาณให้หยุดการทำงาน (Reset) ดังนั้นจึงสามารถแปลแผนภาพฟังก์ชั่นชาร์ต ให้เป็นวงจรฟลิปฟลอปได้ดังรูปที่ 2.7



รูปที่ 2.7 : การแปลงแผนภาพฟังก์ชั่นชาร์ตไปเป็นวงจรฟลิปฟลอป

และจากวงจรฟลิปฟลอปสามารถเขียนสมการบูลลีนของสเต็ป Ei ได้ดังนี้[5]

$$E_i = \overline{E}_{i+1} \cdot (E_{i-1} \cdot T_{i-1} + E_i)$$

Reset Set Latch Reset has priority

รูปที่ 2.8 : สมการบูลลีนของสเต็ป Ei

จากสมการสามารถบรรยายการทำงานของสเต็ป E_i ได้คือ สเต็ป E_i จะทำงานเมื่อขณะนั้นสเต็ป E_{i-1} กำลังแอ็กทีฟอยู่ และเกิดสัญญาณ T_{i-1} ขึ้น โดยที่สเต็ป E_{i+1} จะต้องไม่แอ็กทีฟ จากนั้นสเต็ป E_i จะคงค่า (Hold) ตัวเองไว้ และเมื่อสเต็ป E_{i+1} แอ็กทีฟจะทำให้สเต็ป E_i ถูกรีเซ็ท

---

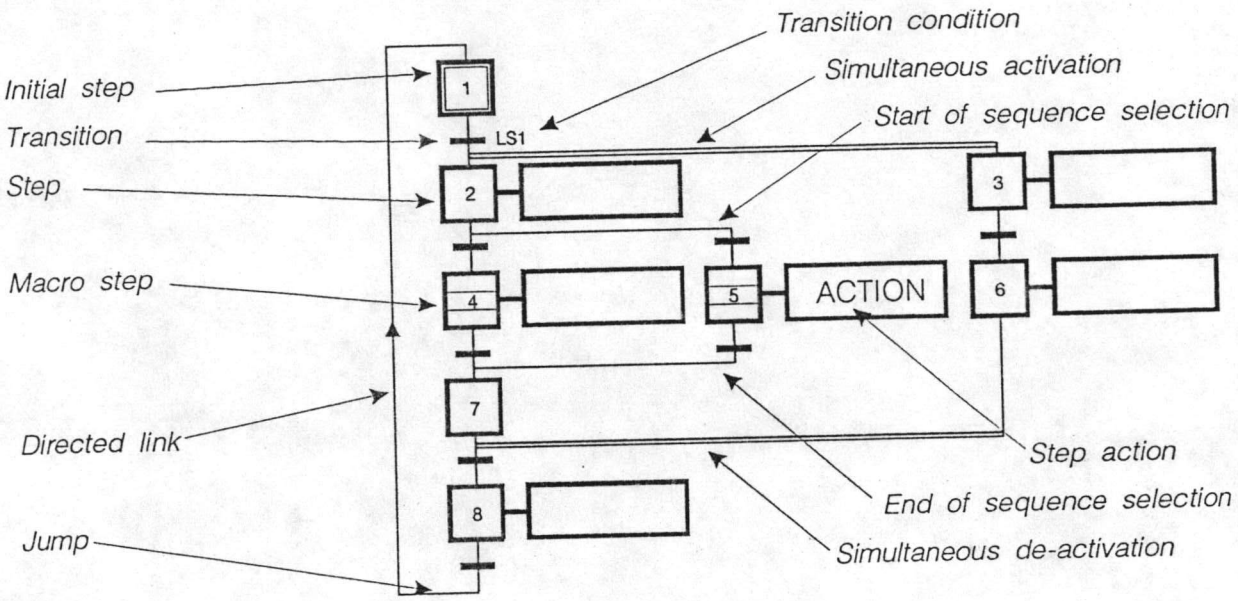
4. SIEMENS. SIEMATIC S5 GRAPH5 Introduction. p. 4.

5. Gilles Michel. Programmable Logic Controller. Architecture and Application. West Sussex, England : John Willey & Sons Ltd., 1990. pp. 261-266.

2.3 หลักภาษา

2.3.1 องค์ประกอบ

ตามมาตรฐาน IEC848[6] ได้กำหนดโครงสร้างและองค์ประกอบต่างๆ ของภาษาฟังก์ชั่นชาร์ตไว้ ดังแสดงในรูปที่ 2.9



รูปที่ 2.9 : องค์ประกอบในภาษาฟังก์ชั่นชาร์ต

Initial Step	:	ใช้ระบุสเต็ปแรกของการควบคุม ซึ่งจะทำให้ทุกครั้งที่เครื่องควบคุมชนิดโปรแกรมได้ทำงานจะมาเริ่มที่สเต็ปนี้เสมอ
Transition	:	ใช้เชื่อมต่อสเต็ปกับสเต็ป เพื่อระบุเงื่อนไขการผ่านสเต็ป
Step	:	ขั้นตอนการทำงานขั้นตอนหนึ่งของการควบคุม สเต็ปจะเก็บคำสั่งควบคุมการทำงาน ในขั้นตอนนั้นไว้
Macro Step	:	เป็นการจัดกลุ่มการทำงานหลาย ๆ ขั้นตอน โดยอ้างอิงด้วย Macro step เพียงสเต็ปเดียว
Directed link	:	แสดงเส้นทางการควบคุมโดยใช้เชื่อมต่อระหว่างสเต็ป และทรานสิชั่น
Jump	:	การเชื่อมต่อสเต็ปที่อยู่ใกล้กัน

---

6. International Electrotechnical Commission. Preparation of Function Chart for Control System. International standard IEC848. 1st ed. Suisse, France : Bureau Central de la Commission Electrotechnique, 1988. pp. 17-49.

Transition condition	:	แสดงเงื่อนไขการผ่านสเต็ป
Simultaneous activation	:	จุดเริ่มต้นของการทำงานพร้อมกัน ใช้สัญลักษณ์เป็นเส้นคู่ แยกออกมาจากด้านใต้ของทรานสิชั่นตัวก่อนหน้า ไปยังเส้นทางการทำงานอื่นๆ
Start of sequence selection	:	จุดเริ่มต้นของการเลือกเส้นทางการทำงาน ใช้สัญลักษณ์เป็นเส้นเดี่ยว แยกออกมาจากด้านบนของทรานสิชั่นตัวก่อนหน้า ไปยังทางเดินต่าง ๆ
Step action	:	การทำงานของขั้นตอนนั้น
End of sequence selection	:	จุดสิ้นสุดของการเลือกเส้นทางการทำงานใช้สัญลักษณ์เป็นเส้นเดี่ยวเชื่อมทางเดิน ต่าง ๆ ที่ต้องการให้สิ้นสุดการเลือกทางเดิน
Simultaneous de-activation	:	จุดสิ้นสุดการทำงานพร้อมกันใช้สัญลักษณ์เป็นเส้นคู่เชื่อมต่อเส้นทางการทำงานต่างๆ ที่ได้ทำงานพร้อมกันมา และต้องการให้สิ้นสุดการทำงานของงานเหล่านั้น

2.3.2 <u>กฎเกณฑ์และข้อกำหนด</u>

1. สเต็ปกับสเต็ปห้ามเชื่อมต่อกันโดยตรง จะต้องคั่นระหว่างกลางด้วยทรานสิชั่น และทรานสิชั่นกับทรานสิชั่นห้ามเชื่อมต่อกันโดยตรง จะต้องคั่นระหว่างกลางด้วยสเต็ป

รูปที่ 2.10 : กฎข้อที่ 1

2. สเต็ปใดที่ต้องการให้เริ่มทำงานเป็นสเต็ปแรกของวงรอบการทำงาน ให้เขียนด้วยสัญลักษณ์ Initial step

3. สถานะของสเต็ป : สเต็ปจะมี 2 สถานะ ได้แก่ สเต็ปที่แอ็กทีฟ (Active step) และสเต็ปที่ไม่แอ็กทีฟ เมื่อใดที่สเต็ปแอ็กทีฟ การทำงานที่ระบุไว้ในสเต็ป จะถูกนำออกไปปฏิบัติ

4. สถานะของทรานสิชั่น : จะมี 2 สถานะคือทรานสิชั่นที่ถูกกระตุ้นและไม่ถูกกระตุ้น

ทรานสิชั่นที่ถูกกระตุ้น (Enabled transition) : ทรานสิชั่นจะถูกกระตุ้น ถ้าหากสเต็ปที่อยู่ก่อนหน้าแอ็กทีฟ

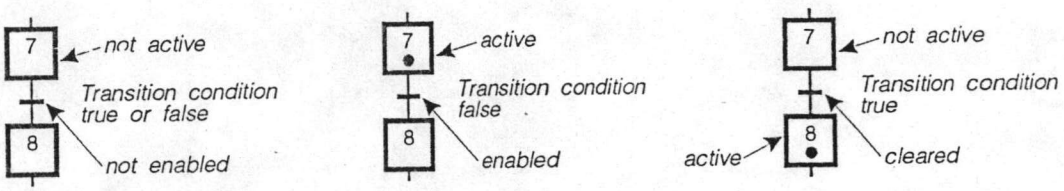
ทรานสิชั่นที่จะถูกเคลียร์ (Clearing of a transition) : ทรานสิชั่นจะถูกเคลียร์ถ้าหากขณะนั้นอยู่ในสถานะที่ถูกกระตุ้น และเงื่อนไขการผ่านสเต็ป ที่ระบุไว้ในทรานสิชั่นนี้เป็นจริง

5. การเคลื่อนของแอ็กทีฟสเต็ป : เมื่อใดที่ทรานสิชั่นถูกเคลียร์ จะก่อให้เกิดผล 3 ประการ ได้แก่

ก. สเต็ปที่อยู่ใต้ทรานสิชั่นจะแอ็กทีฟ

ข. สเต็ปที่อยู่เหนือทรานสิชั่นจะไม่แอ็กทีฟ

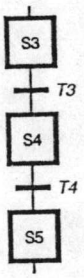
ค. ทรานสิชั่นนั้นจะเปลี่ยนสถานะไปเป็นสถานะไม่ถูกกระตุ้น



รูปที่ 2.11 : การเคลื่อนของแอ็กทีฟสเต็ป

6. โครงสร้างพื้นฐาน

6.1 เส้นทางเดี่ยว (Single sequence) : เป็นการเรียงต่อกันของสเต็ปเป็นอนุกรมซึ่งจะแอ็กทีฟเรียงกันไปตามลำดับ โครงสร้างรูปแบบนี้แต่ละสเต็ปจะตามด้วยทรานสิชั่นเพียง 1 ตัวเท่านั้น และแต่ละทรานสิชั่น จะกระตุ้นสเต็ปให้แอ็กทีฟเพียง 1 สเต็ปเท่านั้น

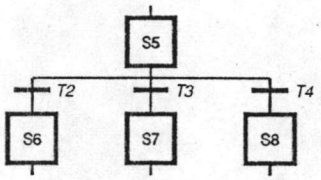


รูปที่ 2.12 : เส้นทางเดี่ยว

6.2 การเลือกเส้นทางการทำงาน (Sequence selection) : เป็นโครงสร้างที่ให้เลือกการทำงาน 1 เส้นทางจากหลาย ๆ เส้นทางที่กำหนดไว้ โดยระบุเงื่อนไขการเลือกเส้นทางไว้ในทรานสิชั่น

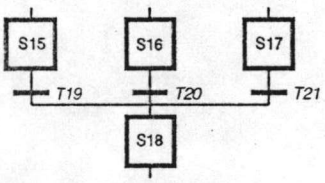
จุดเริ่มต้นของการเลือกเส้นทางการทำงาน : ให้เขียนทรานสิชั่นที่เก็บเงื่อนไข สำหรับการเลือกเส้นทางการทำงานแต่ละเส้นทาง ไว้ใต้เส้นแนวระดับ ที่ลากแยกออกมาจากใต้สเต็ปก่อนหน้า จากรูปที่ 2.13 ถัดจากสเต็ป S5 จะมีสามเส้นทางให้

เส้นทาง ไว้ใต้เส้นแนวระดับ ที่ลากแยกออกมาจากใต้สเต็ปก่อนหน้า จากรูปที่ 2.13 ถัดจากสเต็ป S5 จะมีสามเส้นทางให้เลือกทำงาน จึงเขียนเส้นแนวระดับเชื่อมโยงทางด้านบนของทรานสิชั่นเหล่านั้นเข้าด้วยกัน ถ้าหากขณะนั้นสเต็ป S5 กำลังแอ็กทีฟอยู่ และเงื่อนไขในทรานสิชั่น T2 เป็นจริง สเต็ป S6 จะเป็นสเต็ปต่อไปที่แอ็กทีฟ จะเห็นว่าเกิดการเลือกเส้นทางการทำงานได้



รูปที่ 2.13 : จุดเริ่มการเลือกเส้นทาง

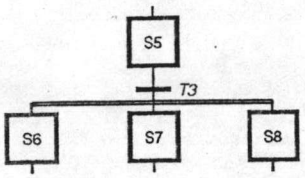
จุดสิ้นสุดของการเลือกเส้นทางการทำงาน : ให้เขียนเส้นแนวระดับเชื่อมโยงจากข้างใต้ของทรานสิชั่น ของแต่ละเส้นทางที่ต้องการให้สิ้นสุดเข้าด้วยกัน จากรูปที่ 2.14 เส้นทางทั้งสามสิ้นสุดพร้อมกัน จึงลากเส้นเชื่อมโยงจากใต้ทรานสิชั่น T19, T20,T21 เข้าด้วยกัน ไม่ว่าจะเกิดการทำงานในเส้นทางไหนเมื่อมาถึงจุดนี้ สเต็ปถัดไปที่จะแอ็กทีฟ จะเป็นสเต็ปเดียวกัน คือสเต็ป S18



รูปที่ 2.14 : จุดสิ้นสุดการเลือกเส้นทาง

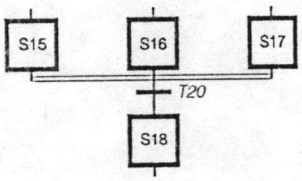
6.3 การทำงานพร้อมกัน (Simultaneous sequence) : เป็นการกระตุ้นให้เกิดการทำงานไปพร้อม ๆ กัน ในหลาย ๆ เส้นทาง โดยใช้ทรานสิชั่น 1 ตัว

จุดเริ่มต้นของการทำงานพร้อมกัน : ให้เขียนเส้นคู่เชื่อมโยงจากเหนือสเต็ปของแต่ละเส้นทางเข้าด้วยกัน โดยเส้นคู่จะต้องอยู่ใต้ทรานสิชั่นที่จะเป็นตัวกระตุ้นให้เกิดการทำงานพร้อมกันของเส้นทางเหล่านั้น จากรูปที่ 2.15 ทรานสิชั่น T3 จะเป็นตัวกระตุ้นให้เกิดการทำงานพร้อมกันใน 3 เส้นทางข้างล่าง ดังนั้นเมื่อการทำงานมาถึง สเต็ป S5 และทรานสิชั่น T3 เป็นจริง สเต็ป S6,S7 และ S8 ก็จะแอ็กทีฟ



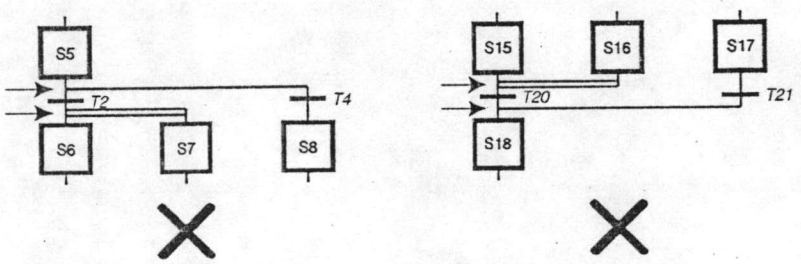
รูปที่ 2.15 : จุดเริ่มทำงานพร้อมกัน

จุดสิ้นสุดของการทำงานพร้อมกัน : ให้เขียนเส้นคู่ลากเชื่อมโยงจากใต้สเต็ปสุดท้ายของแต่ละเส้นทาง เข้ากับด้านบนของทรานสิชั่นตัวถัดไป จากรูปที่ 2.16 การทำงานพร้อมกันในสามเส้นทางจะสิ้นสุดที่สเต็ป S15,S16 และ S17 จึงลากเส้นเชื่อมโยงเข้ากับทรานสิชั่นT20 ดังนั้นทรานสิชั่นT20 จะถูกกระตุ้นก็ต่อเมื่อสเต็ป S15,S16 และ S17 แอ็กทีฟพร้อมกัน (เส้นทางใดทำงานเสร็จก่อนจะต้องรอ) และเมื่อทรานสิชั่น T20 ถูกเคลียร์ ก็จะส่งผลให้สเต็ป S15,S16 และS17 ไม่แอ็กทีฟ อาจกล่าวได้ว่าจุดสิ้นสุดการทำงานพร้อมกันนี้ก็คือการเข้าจังหวะของการทำงานนั่นเอง



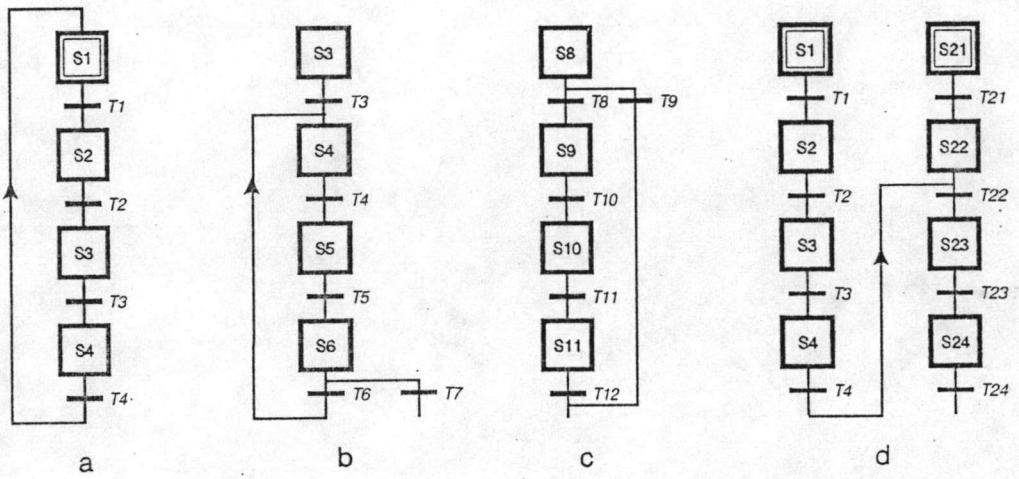
รูปที่ 2.16 : จุดสิ้นสุดการทำงานพร้อมกัน

7. ในการสร้างทางแยกหรือรวมเส้นทางการทำงาน ห้ามเขียนทางแยก 2 ทางแยกซ้อนกัน ดังรูปที่ 2.17



รูปที่ 2.17 : ข้อห้ามในการแยกหรือรวมเส้นทาง

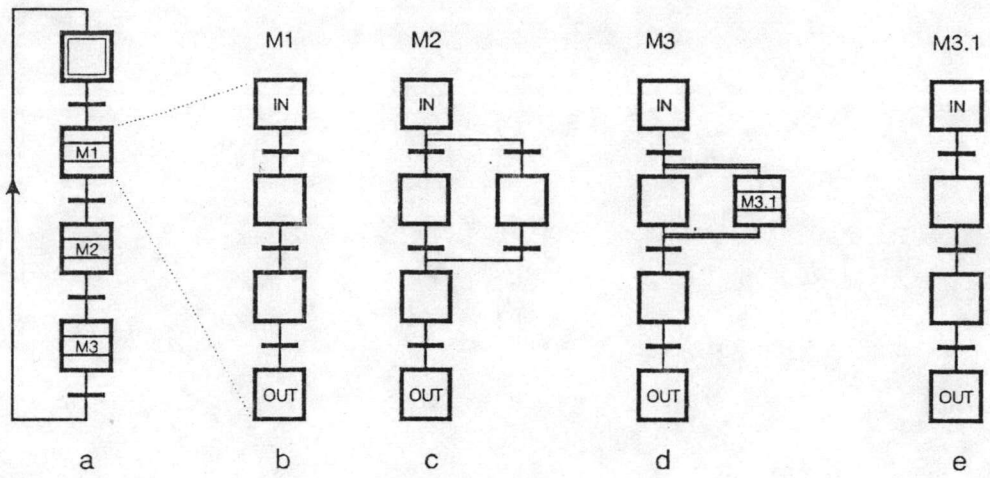
8. การเชื่อมต่อสเต็ประยะไกล (Jump) : ใช้เชื่อมต่อสเต็ปที่อยู่ไกลกัน อาจเพื่อกลับไปเริ่มวงรอบการทำงานใหม่ ดังรูปที่ 2.18a หรือวนรอบกลับไปทำงานซ้ำ ดังรูปที่ 2.18b หรือกระโดดข้ามการทำงานบางช่วงไป ดังรูปที่ 2.18c หรืออาจจะเชื่อมต่อเส้นทางการทำงานไปยังจุดที่ต้องการ ดังรูปที่ 2.18d



รูปที่ 2.18 : การเชื่อมต่อสเต็ประยะไกล

9. แม็คโครสเต็ป (Macro step) : เป็นการจัดกลุ่มการทำงานหลายๆขั้นตอน โดยสามารถอ้างอิงได้ด้วยแม็คโครสเต็ปเพียงสเต็ปเดียว ทั้งนี้เพื่อช่วยให้การเขียนและการอ่านโปรแกรมเข้าใจได้ง่ายขึ้น กล่าวคือ เมื่อเริ่มออกแบบโปรแกรม มองภาพรวมของระบบแบ่งออกเป็นขั้นตอนใหญ่ๆ และเขียนด้วยแม็คโครสเต็ป จากนั้นจึงเขียนโปรแกรมเพื่ออธิบาย แม็คโครสเต็ปเหล่านั้น ซึ่งก็เป็นการออกแบบในลักษณะจากบนลงล่าง (Top-Down design) นั่นเอง จากรูปที่ 2.19a โปรแกรมหลักมี 4 ขั้นตอนการทำงาน โดย 3 ขั้นตอนเป็นแม็คโครสเต็ป ได้แก่ Process 1, Process 2, Process 3 ซึ่งเขียนอธิบายแม็คโครสเต็ปเหล่านั้น ดังรูปที่ 2.19b, 2.19c, 2.19d ตามลำดับ ใน Process3 ใช้แม็คโครสเต็ปอ้างอิงถึง Process 3.1 ซึ่งอธิบายไว้ดังรูป 2.19e

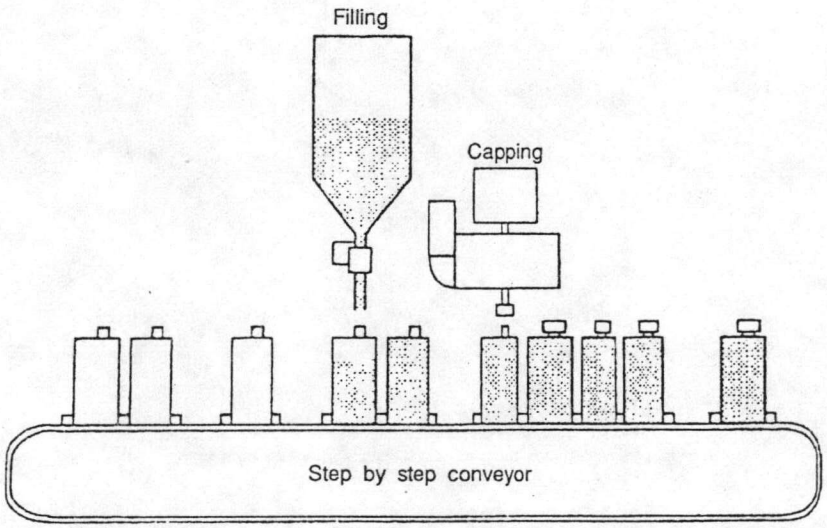
018488



รูปที่ 2.19 : แม็คโครสเต็ป

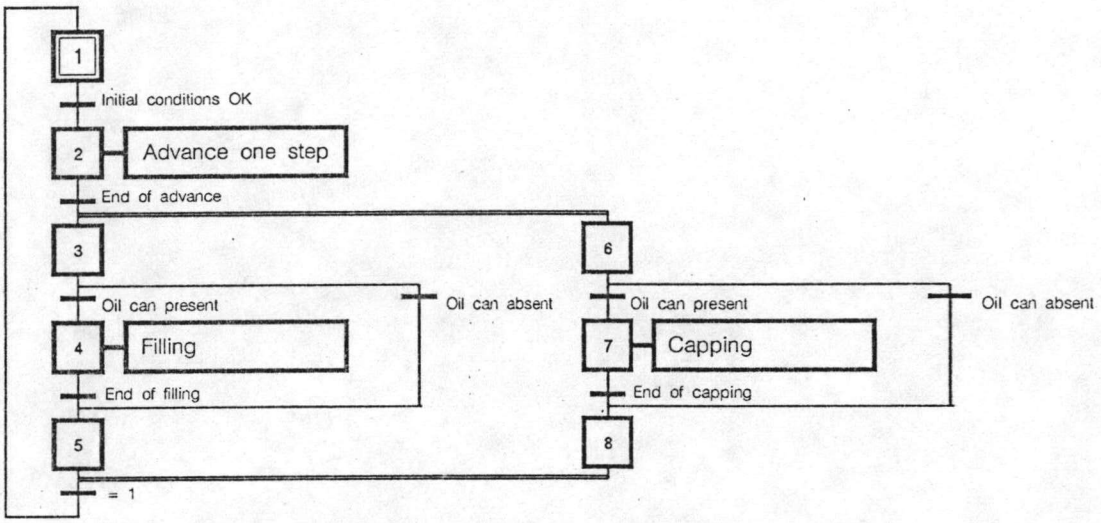
2.4 ตัวอย่าง

2.4.1 เครื่องบรรจุและปิดฝา



รูปที่ 2.20 : เครื่องบรรจุและปิดฝา

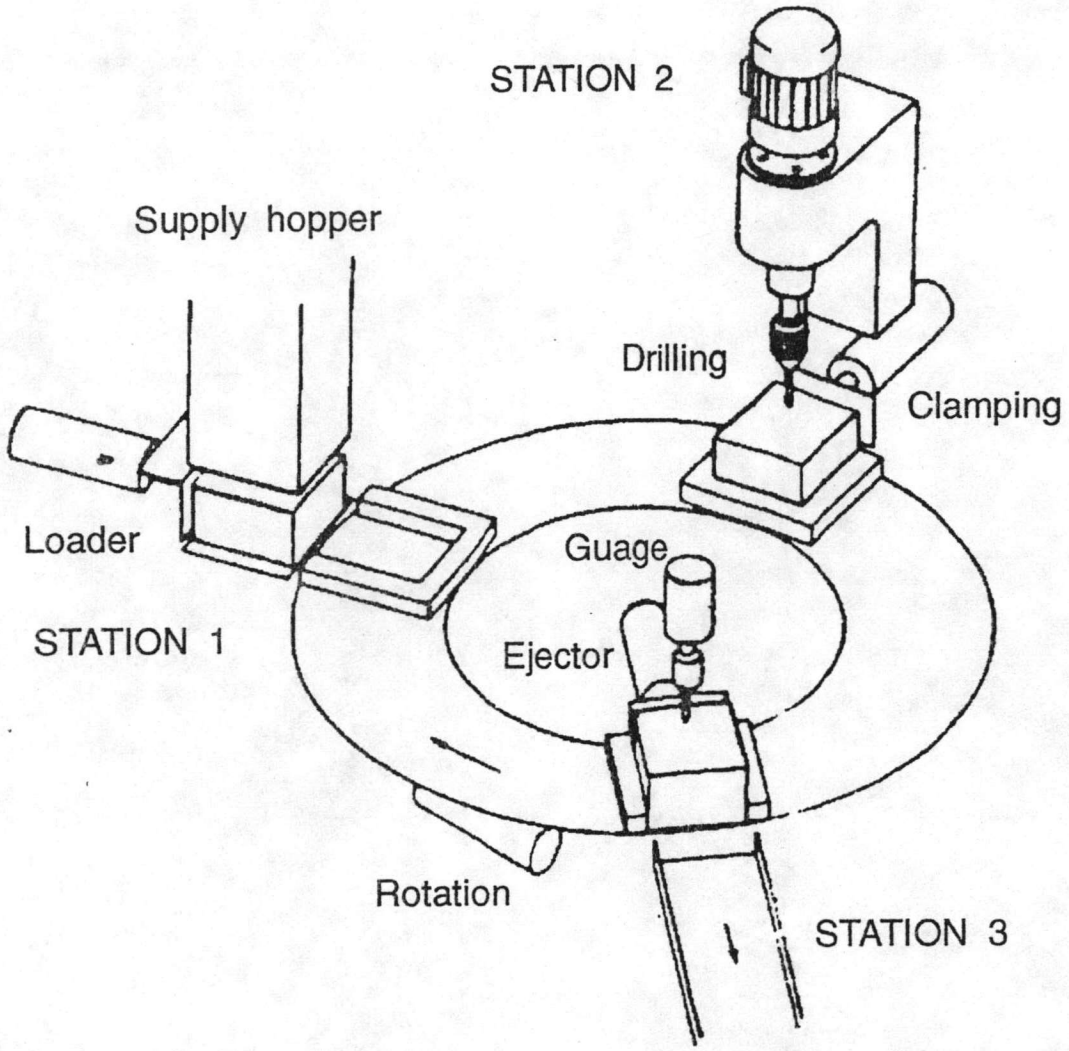
จากภาพแสดงสายการผลิตในส่วนที่จะบรรจุและปิดฝากระป๋องน้ำมัน จะเห็นว่าประกอบด้วย 2 สถานี ได้แก่ สถานีบรรจุน้ำมันและสถานีปิดฝา ซึ่งจะทำงานไปพร้อมๆ กัน โดยมีสายพานเป็นตัวเลื่อนกระป๋องน้ำมันเป็นจังหวะ



รูปที่ 2.21 : ฟังก์ชั่นชาร์ตของเครื่องบรรจุและปิดฝา

การทำงานของเครื่องจักรนี้สามารถเขียนอธิบายได้ดังแผนภาพในรูปที่ 2.21 ที่สเต็ปเริ่มต้นการทำงานจะหยุดรอจนกว่าเงื่อนไขตั้งต้นจะถูกต้องครบถ้วน จากนั้นจึงเลื่อนสายพานไป 1 ช่วง เมื่อสายพานหยุดเลื่อนสถานีทั้งสองจะเริ่มทำงานพร้อมกัน โดยสถานีบรรจุน้ำมันเริ่มต้นที่สเต็ป 3 และสถานีปิดฝาเริ่มต้นที่สเต็ป 6 (ทั้งสเต็ป 3 และสเต็ป 6 เป็นสเต็ปที่ไม่มีการทำงานใด ๆ แต่จะต้องเขียนไว้ มิเช่นนั้นจะผิดกฎในข้อ 7) จากนั้นถ้ามีกระป๋องพร้อมอยู่ ในสเต็ปที่ 4 ก็จะเติมน้ำมันจนเต็ม แล้วไปหยุดรอที่สเต็ป 5 แต่ถ้าไม่มีกระป๋องรออยู่การทำงานก็จะข้ามสเต็ป 4 ไป ส่วนสถานีปิดฝาก็จะปิดฝาในสเต็ป 7 และเคลื่อนไปสเต็ป 8 แต่ถ้าหากไม่มีกระป๋องก็จะข้ามสเต็ป 7 ไป สเต็ปที่ 5 และสเต็ปที่ 8 จะต้องเตรียมไว้ให้สถานีที่ทำงานเร็วกว่ามาหยุดรอ และเมื่อสถานีที่ทำงานช้ากว่าทำมาทันก็จะเกิดการยกเลิกการทำงานพร้อมกัน และผ่านทรานสิชั่นตัวสุดท้าย ซึ่งตั้งให้เงื่อนไขเป็นจริงตลอด (=1) วนกลับไปเริ่มการทำงานในรอบใหม่

2.4.2 แท่นเจาะ



รูปที่ 2.22 : แท่นเจาะ

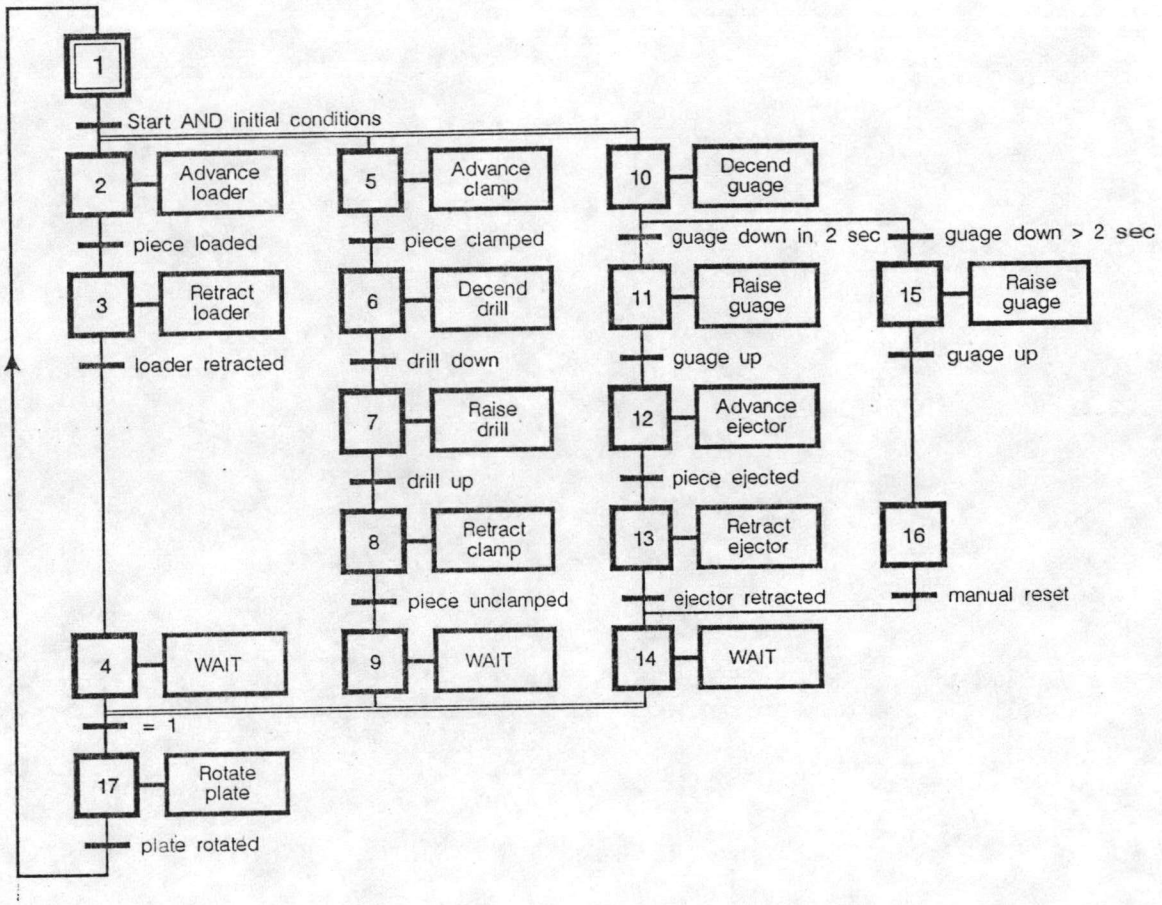
แท่นเจาะนี้จะมี 3 สถานีที่ทำงานไปพร้อม ๆ กัน โดยมีจานหมุนวงกลมช่วยส่งผ่านชิ้นงานไปยังสถานีต่าง ๆ จานจะหมุนครั้งละ 120 องศา แต่ละสถานีจะทำงาน ต่อไปนี้

สถานี 1 : นำชิ้นงานไปวางบนจานหมุน

สถานี 2 : จับชิ้นงานและเจาะรู

สถานี 3 : ทดสอบความลึกของรูที่เจาะ

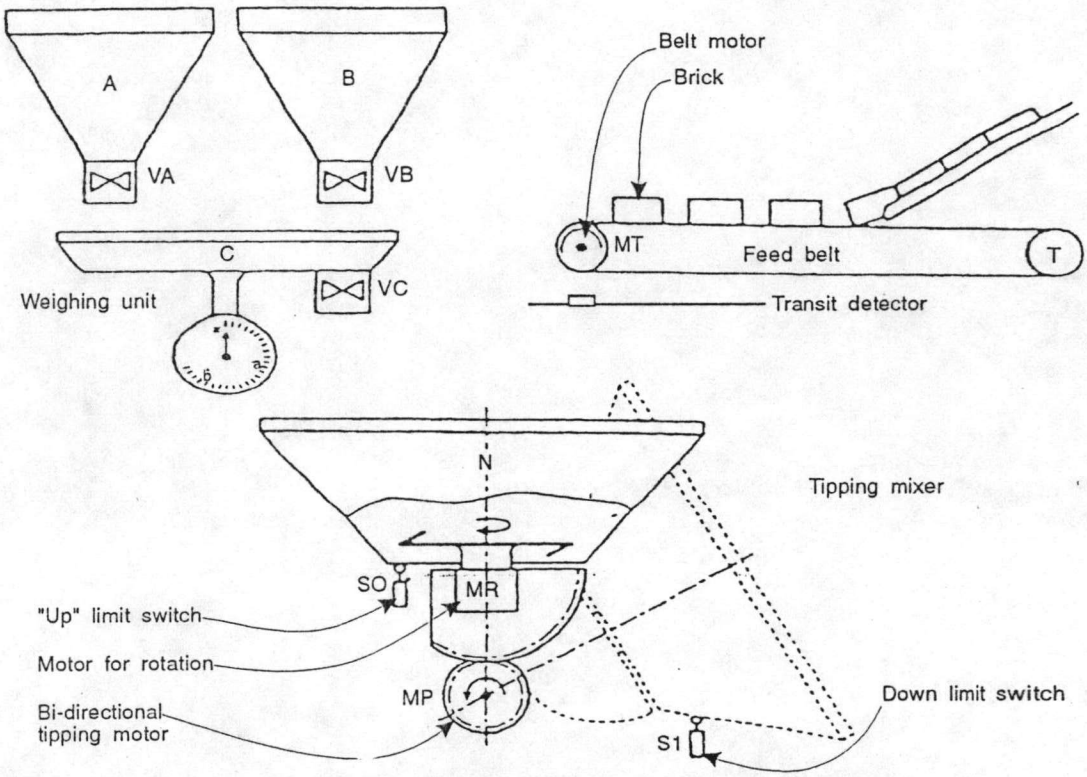
ที่สถานี 3 ถ้าหากวัดไม่ได้ความลึกที่ต้องการ ระบบจะหยุดการทำงาน จนกว่าผู้ดูแลจะนำชิ้นงานนั้นออกไป และกดปุ่มรีเซ็ท



รูปที่ 2.23 : ฟังก์ชั่นชาร์ตของแท่นเจาะ

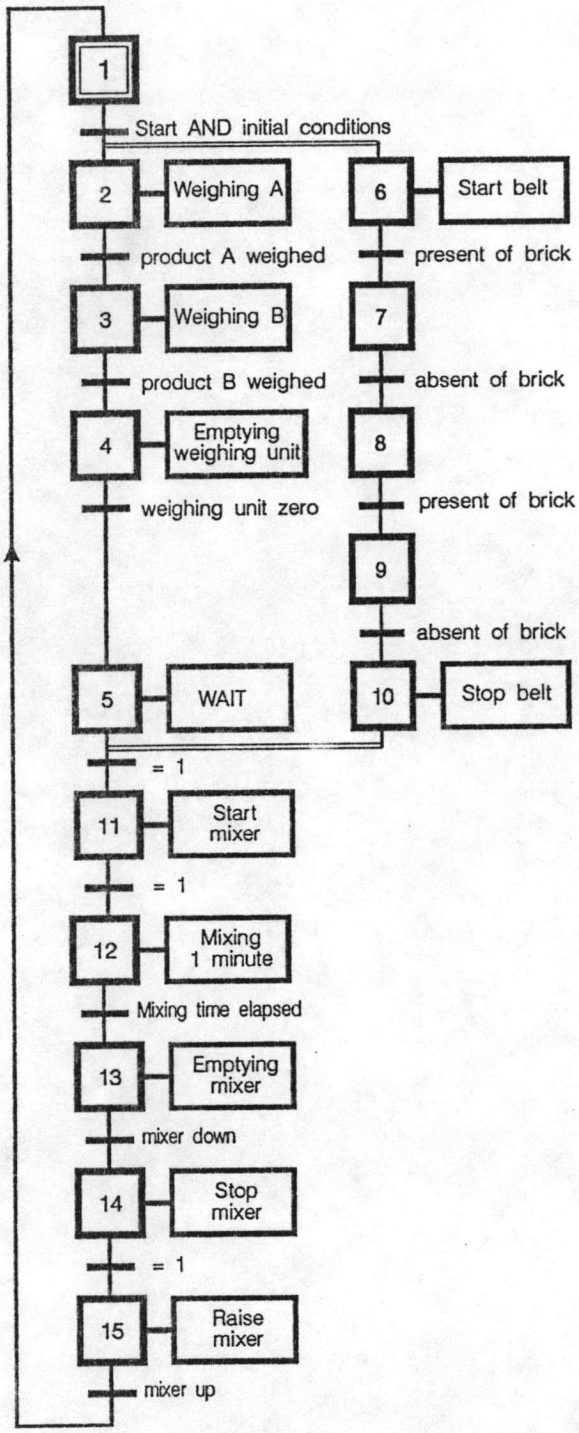
เมื่อมีสัญญาณสตาร์ท จะเกิดการทำงานพร้อมกันใน 3 เส้นทาง ซึ่งจะควบคุมการทำงานในแต่ละสถานี สถานี 1 ถูกควบคุมโดยสเต็ป 2 ถึง 4 สถานี 2 ถูกควบคุมโดยสเต็ป 5 ถึง 9 และสถานี 3 ถูกควบคุมโดยสเต็ป 10 ถึง 16 สถานีใดทำงานเสร็จก่อนก็จะไปรอในสเต็ปท้ายที่เตรียมไว้ในแต่ละเส้นทาง เพื่อการเข้าจังหวะ ที่สถานี 3 จะตรวจสอบความลึกในสเต็ป 10 โดยแหย่เกจ (Depth guage) ลงไปในรู และจับเวลาหากเวลาที่ใช้มากกว่า 2 วินาที แสดงว่ารูที่เจาะไม่ดี ก็จะยกเกจขึ้นและหยุดรอในสเต็ปที่16 เงื่อนไขที่รอก็คือการกดปุ่มรีเซ็ท เมื่อทุกสถานีทำงานมาถึงสเต็ปท้าย (สเต็ป 4,9,14) ก็จะหยุดการทำงานพร้อมกัน จานหมุนก็จะหมุนไป 120 องศา และเริ่มการทำงานรอบใหม่

2.4.3 <u>ระบบชั่งน้ำหนักและผสมอัตโนมัติ</u>



รูปที่ 2.24 : ระบบชั่งน้ำหนักและผสมอัตโนมัติ

เป็นระบบที่จะผสมวัตถุดิบ A, วัตถุดิบ B และอิฐละลายได้ เข้าด้วยกัน โดยวัตถุดิบ A และวัตถุดิบ B จะถูกชั่งน้ำหนักที่ส่วนชั่งน้ำหนัก C (Weighing unit) ตามลำดับ และเมื่อชั่งครบก็จะปล่อยวัตถุดิบ A และ B ลงหม้อผสม ส่วนอิฐละลายได้จะถูกลำเลียง โดยสายพาน และปล่อยลงเครื่องผสมจำนวน 2 ก้อน เมื่อวัตถุดิบทั้งสามพร้อมในหม้อผสม เครื่องผสมก็จะผสมเป็นเวลา 1 นาที แล้วจึงเทของผสมออก

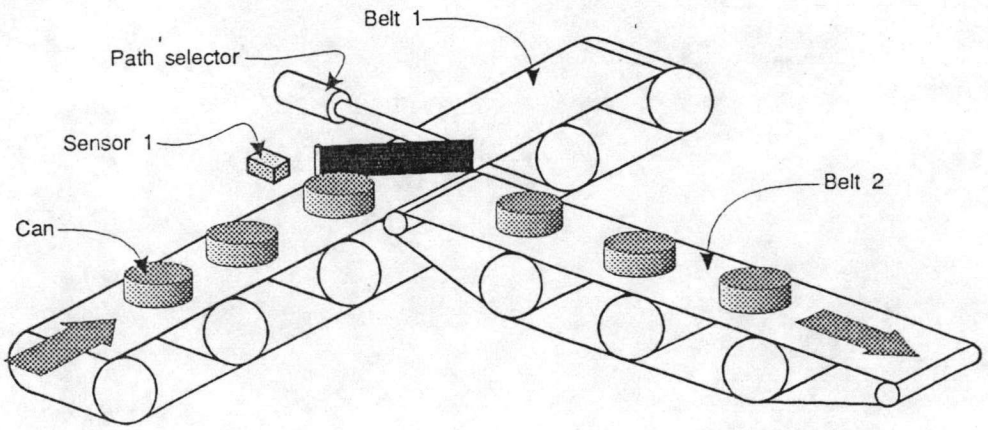


รูปที่ 225 : ฟังก์ชั่นชาร์ตของระบบชั่งน้ำหนักและผสมอัตโนมัติ

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

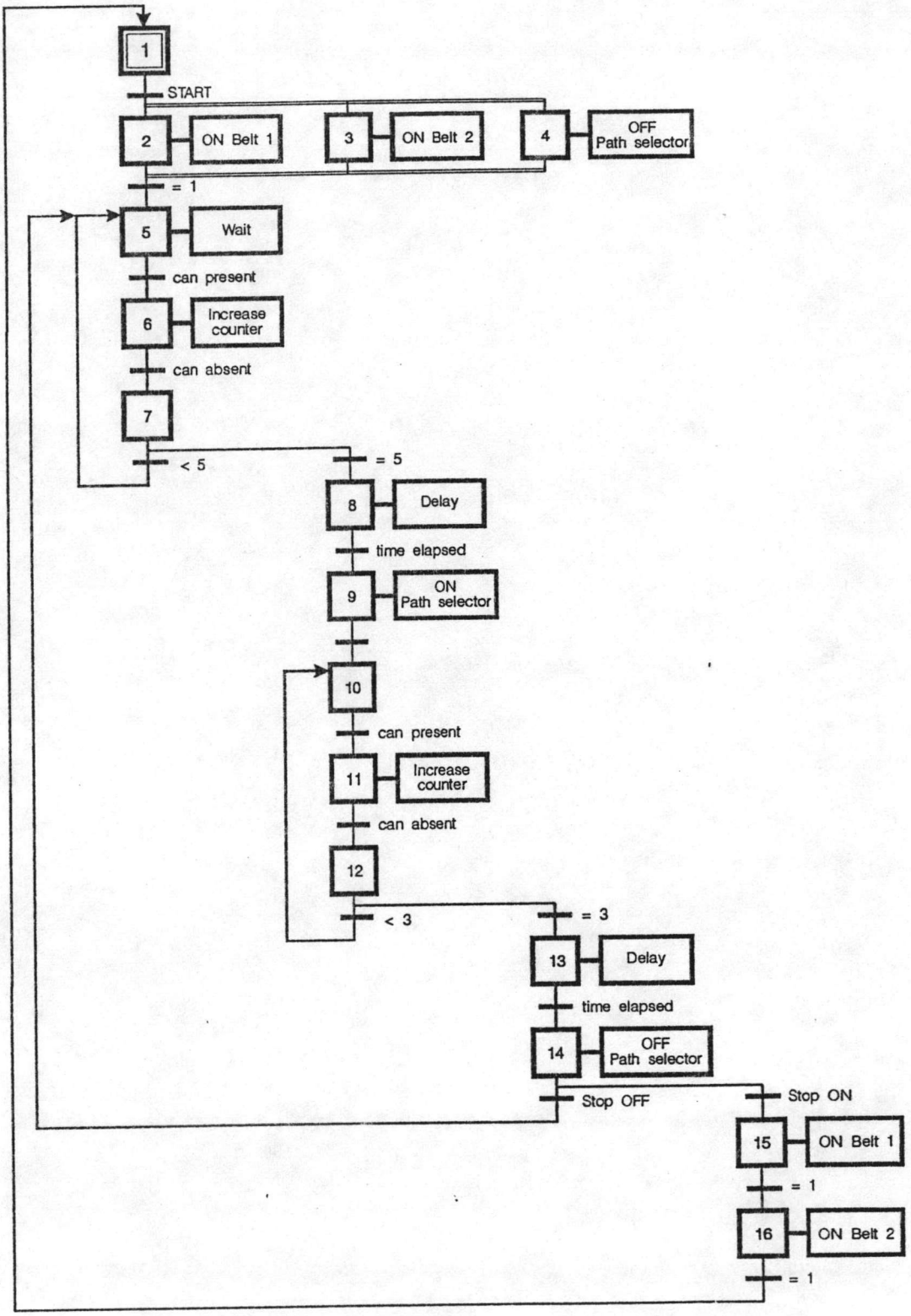
จากฟังก์ชั่นชาร์ตเมื่อกดปุ่มสตาร์ท จะเกิดการทำงานพร้อมกันใน 2 ส่วน ได้แก่ ส่วนชั่งน้ำหนักและสายพานลำเลียงอิฐ ที่ส่วนชั่งน้ำหนัก จะชั่งน้ำหนักวัตถุดิบ A ก่อนแล้วจึงชั่งน้ำหนักวัตถุดิบ B จากนั้นจะเปิดวาล์ว VC ปล่อยวัตถุดิบ A และ B ลงหม้อผสม หากในส่วนนี้ทำงานเสร็จก่อนส่วนสายพาน ก็จะหยุดรอในสเต็ปที่ 5 ที่สายพานจะมีตัวตรวจจับอิฐ (Transit detector) โดยหลังจากเดินสายพานในสเต็ปที่ 6 แล้ว จะรอสัญญาณการวิ่งผ่านของอิฐ และรอให้สัญญาณหมดไป แสดงว่าอิฐวิ่งผ่านไปหนึ่งก้อน ทำเช่นนี้อีกหนึ่งครั้ง อิฐก็จะผ่านไปอีกหนึ่งก้อน งานในช่วงนี้เป็นการรอการเกิดของลำดับสัญญาณ จึงไม่มีการทำงานใดๆ ในสเต็ป 7,8,9 เมื่ออิฐผ่านครบ 2 ก้อน ก็จะสั่งให้สายพานหยุดเดิน จากนั้นจะเป็นการทำงานของเครื่องผสม ส่วนชั่งน้ำหนักและสายพานจะหยุดทำงาน สเต็ป11 จะเดินเครื่องผสม และไปทำงานในสเต็ป 12 ทันที (ทรานสิชั่นเป็นจริงตลอด) จะจับเวลา 1 นาที จากนั้นจะเทของผสมออก เมื่อหม้อผสมเอียงลงจนถึงสวิตช์ S1 สเต็ป 14 ก็จะทำงาน กล่าวคือ จะสั่งให้เครื่องผสมหยุดหมุน และยกหม้อผสมขึ้น จนถึงสวิตช์ S0 ก็จะวนกลับไปทำงานรอบใหม่

2.4.4 <u>สายพานลำเลียง</u>



รูปที่ 2.26 : สายพานลำเลียง

จากรูปแสดงการเชื่อมต่อของสายพาน 2 เส้น โดยมีตัวกำหนดเส้นทาง (Path selector) เป็นตัวกำหนดเส้นทางการเคลื่อนที่ของกระป๋อง จังหวะแรกจะให้กระป๋องผ่านไปตามสายพาน1 (Belt 1) ตลอดสายพาน จำนวน 5 กระป๋อง จากนั้นจะเปลี่ยนเส้นทางให้กระป๋องเลี้ยวไปใช้สายพาน 2 จำนวน 3 กระป๋อง สลับไปมา จนกว่าจะเปิดปุ่ม STOP



รูปที่ 2.27 : ฟังก์ชั่นชาร์ตของสายพานลำเลียง

จากฟังก์ชั่นชาร์ต เมื่อกดปุ่มสตาร์ท สายพาน 1 และสายพาน 2 จะเริ่มเดิน ตัวกำหนดเส้นทางจะเปิดทางให้กระป๋องเคลื่อนผ่านไปตามสายพาน1 (สเต็ป 4) และหยุดรอที่สเต็ป 5 จนกว่าจะมีกระป๋องวิ่งมาถึงตัวตรวจจับ สเต็ป 6 ก็จะนับและเมื่อกระป๋องเคลื่อนผ่านตัวตรวจจับไป สเต็ป 7 จะแอ็กทีฟ ซึ่งไม่มีการทำงานใดๆ แต่จะเลือกทางเดินโดยการเปรียบเทียบค่าในตัวนับ (Counter) หากน้อยกว่า 5 ก็จะวนกลับไปทำสเต็ป 5 แต่ถ้าหากเท่ากับ5 สเต็ป 8 ก็จะแอ็กทีฟ โดยการหยุดรอชั่วขณะหนึ่ง เพื่อให้กระป๋องผ่านตัวกำหนดเส้นทางไปก่อน จากนั้นสเต็ป9 จะสั่งให้ตัวกำหนดเส้นทางปิดทางเดิน เพื่อให้กระป๋องผ่านไปทางสายพาน 2 แทน การทำงานจะหยุดรอที่สเต็ป 10 จนกว่าจะมีกระป๋องเคลื่อนผ่านตัวตรวจจับ ตัวนับก็จะนับและเปรียบเทียบค่า หากน้อยกว่า 3 ก็จะกลับไปทำสเต็ป 10 แต่ถ้าหากเท่ากับ 3 ก็จะหยุดชั่วขณะ(สเต็ป 13) เพื่อให้กระป๋องผ่านตัวกำหนดเส้นทาง แล้วจึงสั่งให้ตัวกำหนดเส้นทางเปิดทางเดินไปตามสายพาน 1 หากปุ่ม STOP ไม่ถูกเปิด การทำงานจะวนกลับไปที่สเต็ป 5 แต่ถ้าหากเปิดปุ่ม STOP สเต็ป 15 และ 16 จะสั่งให้สายพานทั้งสองหยุดเดิน และวนกลับไปที่สเต็ปเริ่มต้น