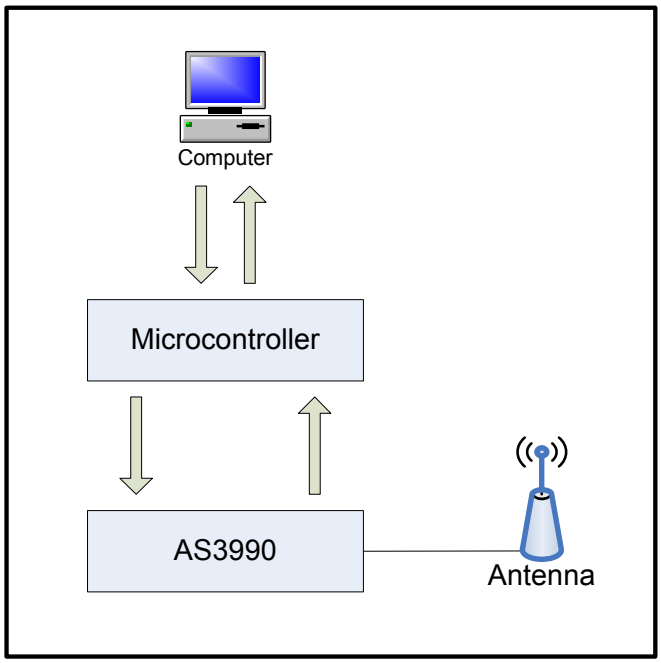
# บทที่ 3

# การพัฒนาระบบป้องกันการชนกันของข้อมูลในระบบอาร์เอฟไอดี

ปัญหาของระบบป้องกันการชนกันของข้อมูลในระบบอาร์เอฟไอดี คือ การที่มีป้ายมากกว่า 1 ป้ายเข้ามาในระยะการทำงานของเครื่องอ่านแล้วทำการส่งข้อมูลออกมาในเวลาเดียวกัน จึงเกิดการชนกันของข้อมูล ในปัญหานี้ถ้าเราใช้การป้องกันการชนกันของข้อมูลแบบ SDMA, FDMA และ CDMA จะต้องทำการพัฒนา ฮาร์ดแวร์ ของป้าย ซึ่งทำให้ราคาของป้ายนั้นมีราคาสูงขึ้น ดังนั้นจึงสมควรใช้การป้องกันการชนกันของข้อมูลแบบ TDMA ซึ่งใช้วิธีการแบ่งช่วงเวลาให้ป้ายตอบสนองมาในช่วงเวลาหนึ่งๆเพียง 1 ป้ายซึ่งพัฒนาในส่วนของฮาร์ดแวร์เพียงเล็กน้อย งานวิจัยนี้จึงมุ่งเน้นพัฒนาในส่วนของวิธีการป้องกันการชนกันของข้อมูลเพื่อแก้ไขปัญหาดังกล่าว ซึ่งวิธีทำการวิจัยนี้จะเป็นการทดสอบและเปรียบเทียบประสิทธิภาพของเทคนิคที่นำเสนอในการแก้ไขปัญหาการชนกันของข้อมูล โดยงานวิจัยนี้จะเน้น การป้องกันการชนกันของข้อมูลแบบ TDMA ซึ่งแบ่งออกเป็น 2 วิธีหลักๆคือ BT และ Aloha โดยมีวิธีการทำวิจัยแบ่งออกเป็นสองส่วน คือ มาตรฐานและเครื่องอ่านที่ใช้ทำวิจัย, การจำลองการทำงานของอัลกอริทึมป้องกันการชนกันของข้อมูล, การสร้างอัลกอริทึมป้องกันการชนกันของข้อมูลในเครื่องอ่าน และ โปรแกรมควบคุมการทำงานของเครื่องอ่าน

## 3.1 มาตรฐานและเครื่องอ่านที่ใช้ทำวิจัย

มาตรฐานที่งานวิจัยนี้ได้นำมาวิจัยเป็นมาตรฐาน ISO 18000 - 6 ประเภท A, B และ C เนื่องจากเป็นมาตฐานที่ใช้ความถี่สูงยิ่ง ทำให้ระยะการอ่านของเครื่องอ่านนั้นมีระยะที่ไกลขึ้น ซึ่งทำให้รองรับป้ายจำนวนมากๆได้ เครื่องอ่านที่นำมาใช้ทำวิจัยนั้น เป็นเครื่องอ่านย่านความถี่สูงยิ่ง 860 ถึง 960 MHz ซึ่งเป็น บอร์ดพัฒนาใช้ชิป AS3990 ของบริษัท Austriamicrosystems ซึ่งใช้ไมโครคอนโทรลเลอร์ตระกูล MSP เบอร์ MSP430F156 เป็นตัวควบคุมการทำงานของเครื่องอ่าน



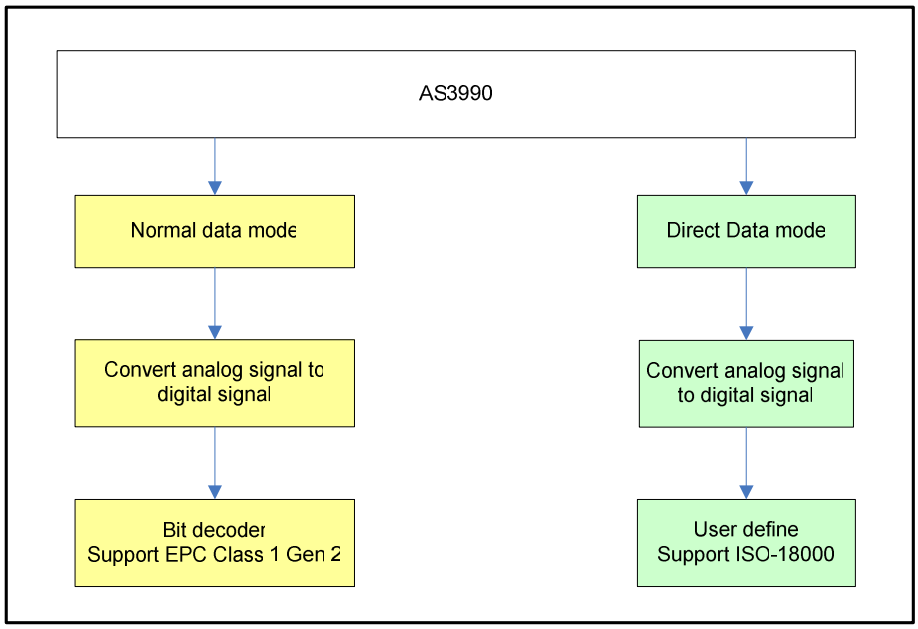

**รูปที่** 3.1 การเชื่อมต่อของระบบเครื่องอ่าน

รูปที่ 3.1 การเชื่อมต่อของระบบเครื่องอ่านโดย สายอากาศจะทำหน้าที่รับส่งสัญญาณความถี่วิทยุระหว่างเครื่องอ่านกับป้าย โดยสัญญาณต่างๆจะถูกเข้ารหัสและถอดรหัสที่ชิป AS3990 ซึ่งชิป AS3990 จะส่งข้อมูลให้กับไมโครคอนโทรลเลอร์ประมวลผลข้อมูล จากนั้นจึงส่งไปให้กับผู้ใช้งานผ่านพอร์ตยูเอสบี (USB: universal serial bus) ซึ่งเป็นการทำงานเสมือนพอร์ตอนุกรม

การทำงานของ AS3990 นั้นจะแบ่งเป็น 2 ประเภท คือ Normal data mode และ Direct data mode ในการทำงาน 2 ประเภทนี้จะมีความแตกต่างกันคือ

### 3.1.1 โหมดการทำงานแบบเข้าถึงข้อมูลแบบปกติ (normal data mode)

สนับสนุนการทำงานของมาตรฐาน ISO 18000-6 C หรือ EPC Class 1 Gen 2 ในการทำงานประเภทนี้จะมีการเข้ารหัสและถอดรหัสตามมาตรฐาน ISO 18000-6 ประเภท C ดังนั้นผู้ใช้งานในมาตรฐานนี้สามารถเขียนโปรแกรมให้เครื่องอ่านทำงานได้ โดยไม่ต้องเขียนโปรแกรมจัดการกลุ่มข้อมูลระดับล่างของ คำสั่งของเครื่องอ่าน และ คำตอบสนองจากป้าย




**รูปที่** 3.2 ประเภทการทำงานของเครื่องอ่าน

**รูปที่** 3.3 เครื่องอ่านบัตรอาร์เอฟไอดี

3.1.2 โหมดการทำงานแบบเข้าถึงข้อมูลโดยตรง (direct data mode)

การทำงานประเภทนี้จะทำงานโดยผู้ใช้งานจะต้องทำการ ออกแบบการส่งและรับข้อมูลด้วยตัวเอง โดยจะมีเครื่องมือในการเปลี่ยนสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล จากนั้นผู้ใช้งานจะต้องเขียนโปรแกรมเพื่อทำงานในการนำข้อมูลไปถอดรหัส จึงจะนำข้อมูลไปใช้งานได้

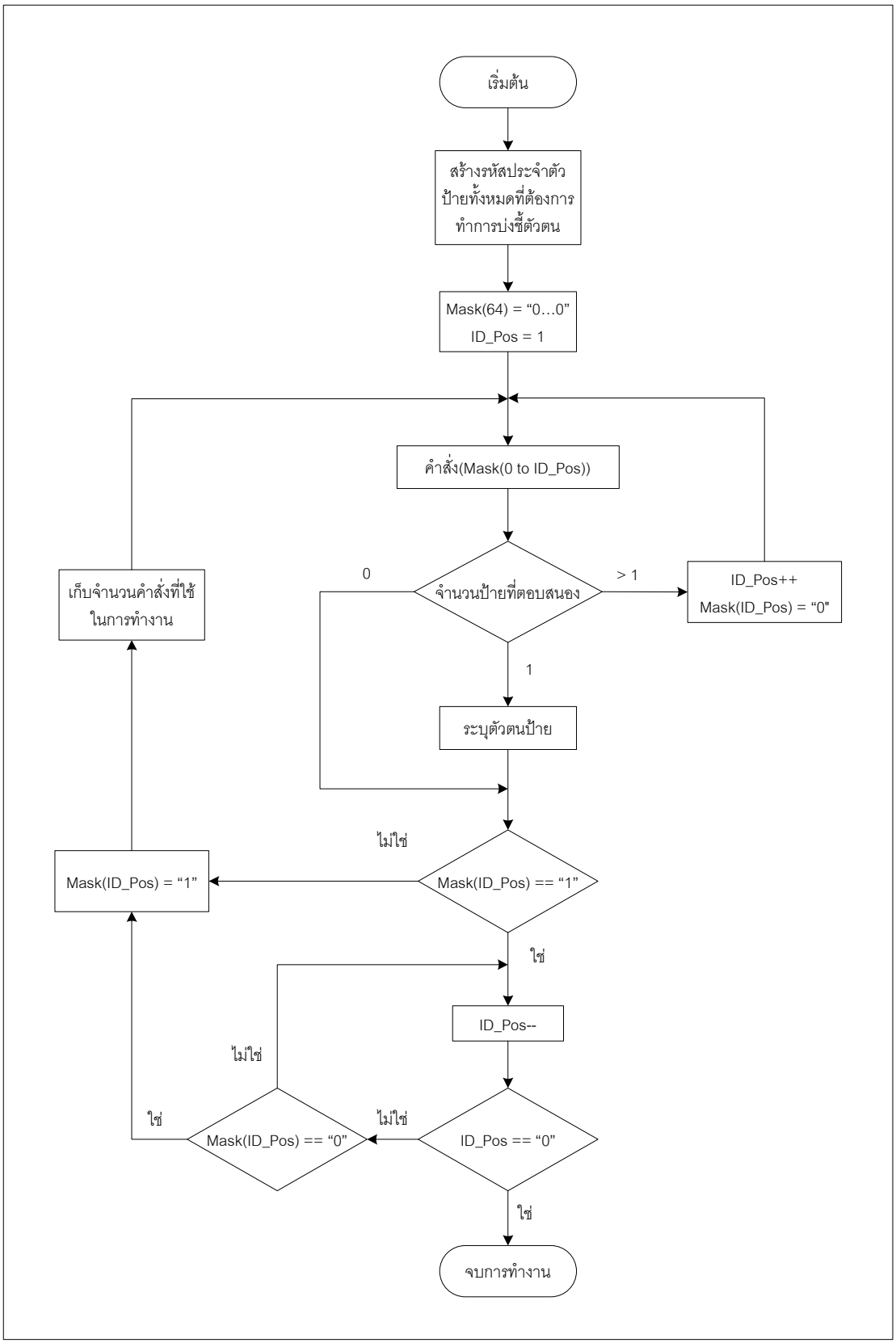
## 3.2 การจำลองการทำงานของอัลกอริทึมป้องกันการชนกันของข้อมูล

การจำลองการทำงานของอัลกอริทึมป้องกันการชนกันของข้อมูลนั้น ผู้วิจัยได้ทำการจำลองการทำงานของ BT, FSA, DFSA, A New Anti-Collision Based on A-Priori Information และ A Novel Anti – collision Algorithm for High - Density RFID Tags โดยใช้โปรแกรม Scilab ในการจำลองการทำงานโดยทำการเก็บจำนวนคำสั่งที่ใช้, จำนวนข้อมูลที่ทำการรับส่ง และ เวลาที่ให้ในการทำงานมาทำการเปรียบเทียบประสิทธิภาพในการทำงานของแต่ละอัลกอริทึม

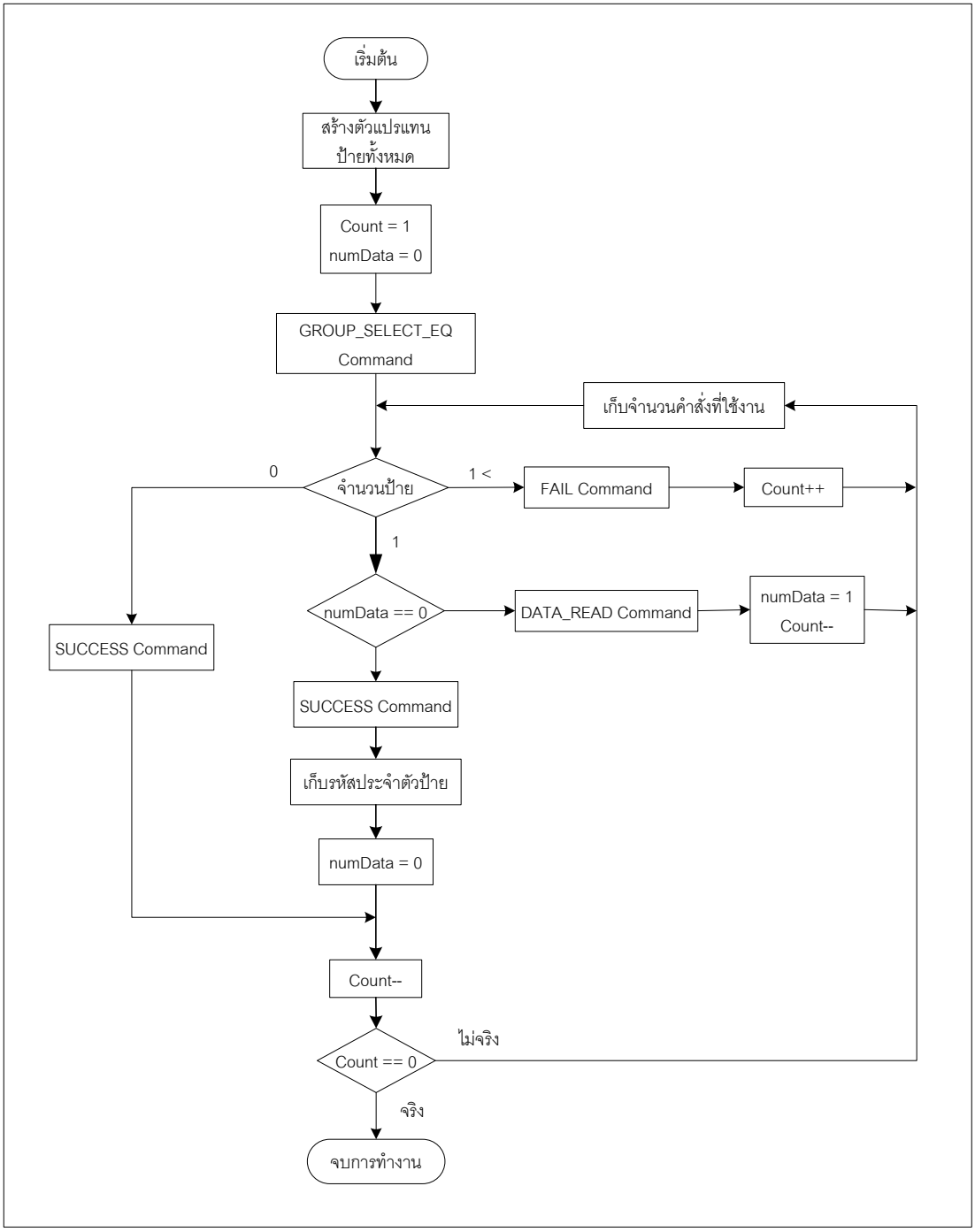
3.2.1 อัลกอริทึมต้นไม้แบบทวิภาค

ในหลักการทำงานของอัลกอริทึมต้นไม้แบบทวิภาค ผู้วิจัยได้อธิบายหลักการทำงานเบื้องต้นไปแล้วในบทที่ 2 ในส่วนนี้จะอธิบายถึงการจำลองการทำงานของอัลกอริทึม ซึ่งมีการทำงานเป็นไปตามรูปที่ 3.4 เมื่อเริ่มต้นการทำงานจะทำการสร้างรหัสประจำตัวป้ายจำนวน 64 บิต โดยในการสร้างรหัสประจำตัวป้ายจะใช้วิธีสุ่มค่ารหัสประจำตัวด้วยความน่าจะเป็นที่มีรูป แบบ ยูนิฟอร์ม ซึ่ง มีความน่าจะเป็นในการสุ่มรหัสจะเท่ากันทุกรหัส

จากนั้นจึงนำมาเปรียบเทียบกับกลุ่มของรหัสของป้ายที่ ทำการสุ่มก่อนหน้านี้ ถ้าไม่ซ้ำกันจะทำการเก็บรหัสไว้ในฐานข้อมูล แต่ถ้าซ้ำกันจะทำการสุ่มรหัสประจำตัวป้ายใหม่จากนั้นจึงเริ่มกระบวนการป้องกันการชนกันของข้อมูลแบบ อัลกอริทึมต้นไม้แบบทวิภาคโดยเริ่มจากการส่งคำสั่งไปให้กับป้ายและตรวจนับป้ายที่ตอบสนองจากนั้นจึงทำการตัดสินใจในการเพิ่มลดหรือเปลี่ยนส่วนหน้าของรหัสป้ายในคำสั่ง และ ทำการบันทึกจำนวนคำสั่งที่ใช้งาน เพื่อนำมาเปรียบเทียบประสิทธิภาพ




**รูปที่** 3.4 แผนผังการจำลองการทำงานของอัลกอริทึมต้นไม้แบบทวิภาค




**รูปที่ 3.5** แผนผังการจำลองการทำงานของ ISO 18000-6 ประเภท B

### 3.2.2 อัลกอริทึมต้นไม้แบบทวิภาคที่ใช้ใน ISO 18000-6 ประเภท B

ในการจำลองการทำงานของการป้องกันการชนกันของข้อมูลตามมาตรฐาน ISO 18000-6 ประเภท B ตามรูปที่ 3.5 จะเริ่มจากการสร้างตัวแปรนับ 8 บิตภายในตัวป้ายทุกป้ายเพื่อ

ใช้สำหรับเก็บลำดับการตอบสนองของป้ายและสร้างตัวแปรนับภายในเครื่องอ่าน (Count) เพื่อตรวจสอบจำนวนกลุ่มทั้งหมดที่แบ่ง ซึ่งรายละเอียดสามารถดูได้จากบทที่ 2 จากนั้นจึงเริ่มต้นด้วยการส่งคำสั่ง GROUP_SELECT_EQ เพื่อเลือกป้ายที่จะทำการป้องกันการชนกันของข้อมูล จากนั้นจึงพิจารณาถึงจำนวนป้ายที่ตอบสนองกลับมา โดยเข้าไปค้นหาป้ายที่มีค่าตัวแปรนับเท่ากับ 0 ในตัวแปรที่เก็บค่าตัวแปรนับของแต่ละป้าย ซึ่งมีความเป็นไปได้ 3 แบบดังนี้

1. ไม่มีป้ายตอบสนอง(Idle)

   โปรแกรมจะทำการส่งคำสั่ง SUCCESS เพื่อทำการลดค่าตัวแปรนับในป้ายลง 1 ค่า และ ลดค่าตัวแปรนับในเครื่องอ่านลง 1 ค่า

2. มีป้ายตอบสนอง 1 ป้าย(Success)

   โปรแกรมจะทำการส่งคำสั่ง DATA_READ พร้อมกับรหัสประจำตัวป้ายที่ได้รับมา เพื่อย้าย ป้ายไปอยู่ในสถานะการแลกเปลี่ยนข้อมูลจากนั้นป้ายจะตอบสนองกับโปรแกรมด้วยรหัสประจำตัวป้าย เมื่อโปรแกรมได้รับรหัสประจำตัวป้ายตรงกับที่ส่งไปในคำสั่ง DATA_READ ก็จะทำการบ่งชี้ตัวตนป้ายนั้นๆ และ ส่งคำสั่ง SUCCESS เพื่อทำการลดค่าตัวแปรนับในป้ายอีก 1 ค่า และ ลดค่าตัวแปรนับในเครื่องอ่านลง 1 ค่า

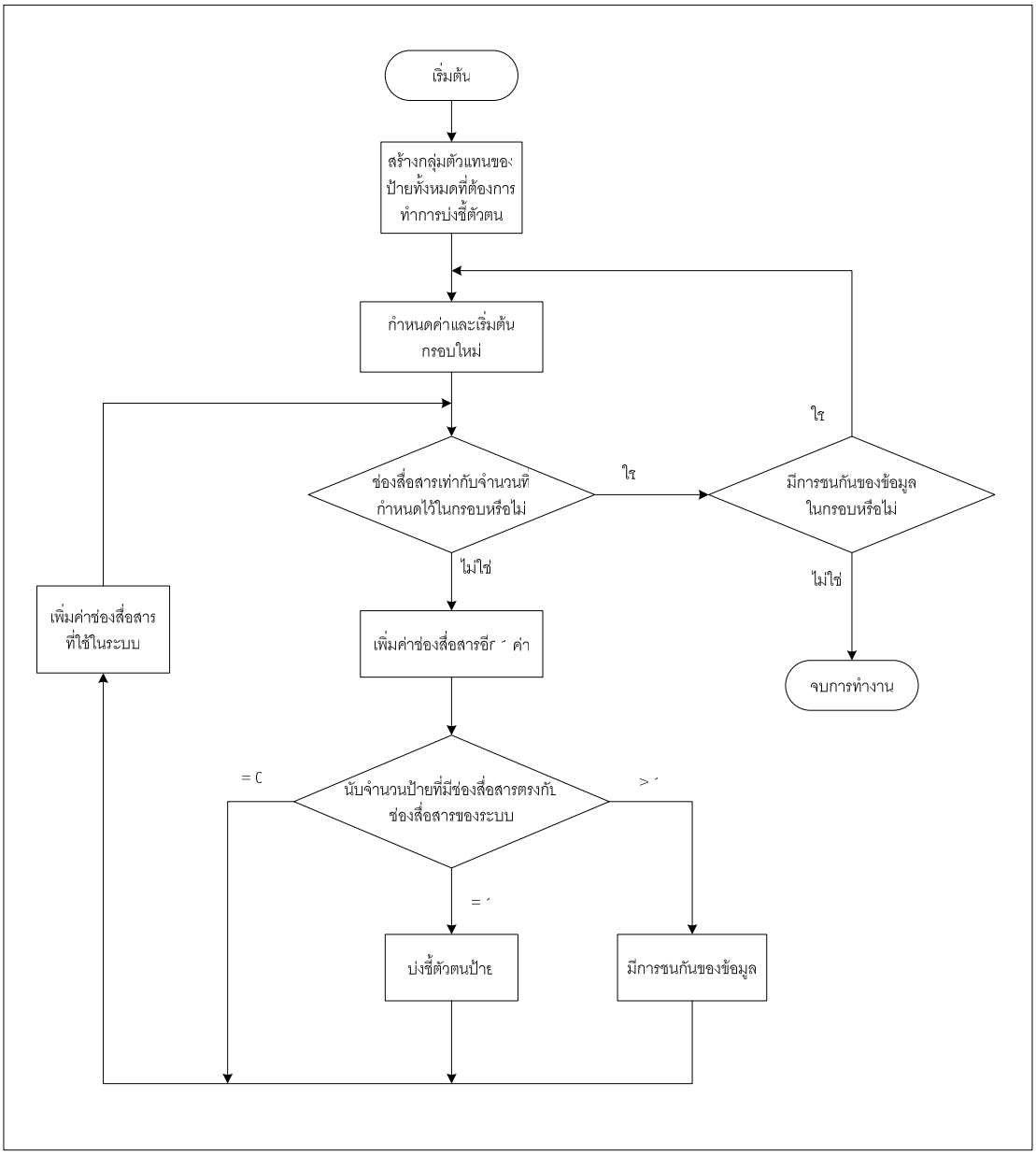
3. มีป้ายตอบสนองมากกว่า 1 ป้าย(Collision)

   โปรแกรมจะทำการส่งคำสั่ง FAIL เพื่อทำการเพิ่มค่าตัวแปรนับในป้าย 1 ค่า และลดค่าตัวแปรนับในเครื่องอ่านลง 1 ค่า

โปรแกรมจะทำงานตามคำสั่งด้านบนไปจนกว่าค่าตัวแปรนับที่เครื่องอ่านจะเท่ากับ 0 จึงหยุดการทำงาน จากนั้นโปรแกรมจะทำการเก็บจำนวนคำสั่งที่ใช้งานในการป้องกันการชนกันของข้อมูลเพื่อนำมาเปรียบเทียบกับการป้องกันการชนกันของข้อมูลแบบอื่นๆ

### 3.2.3 อัลกอริทึม Framed Slotted Aloha

ในการจำลองการทำงานของการป้องกันการชนกันของข้อมูลแบบ FSA นั้นผู้วิจัยได้ทำการเขียนโปรแกรมจำลองการทำงานโดยเริ่มต้นโดยสร้างตัวแปรเพื่อเก็บค่าตัวแปรที่สำคัญของป้ายทั้งหมดในระยะการทำงานเช่น จำนวนช่องสื่อสารที่สุ่มได้ จากนั้นจึงเริ่มกระบวนการป้องกันการชนกันของข้อมูล โดยส่งคำสั่งให้กับป้ายพร้อมกับจำนวนช่องสื่อสารที่ใช้อยู่ถ้า ป้ายใดมีจำนวนช่องสื่อสารที่สุมได้เท่ากับจำนวนช่องสื่อสารที่โปรแกรมส่งไปก็จะทำการตอบสนองกับมาดังรูปที่ 3.6 ซึ่งเหตุการณ์ที่อาจเกิดขึ้นได้ มี 3 กรณีดังต่อไปนี้




**รูปที่** 3.6 แผนผังการจำลองการทำงานของ FSA

1. ไม่มีป้ายตอบสนอง

โปรแกรมจะทำส่งคำสั่งพร้อมทั้งช่องสื่อสารถัดไปให้กับป้าย

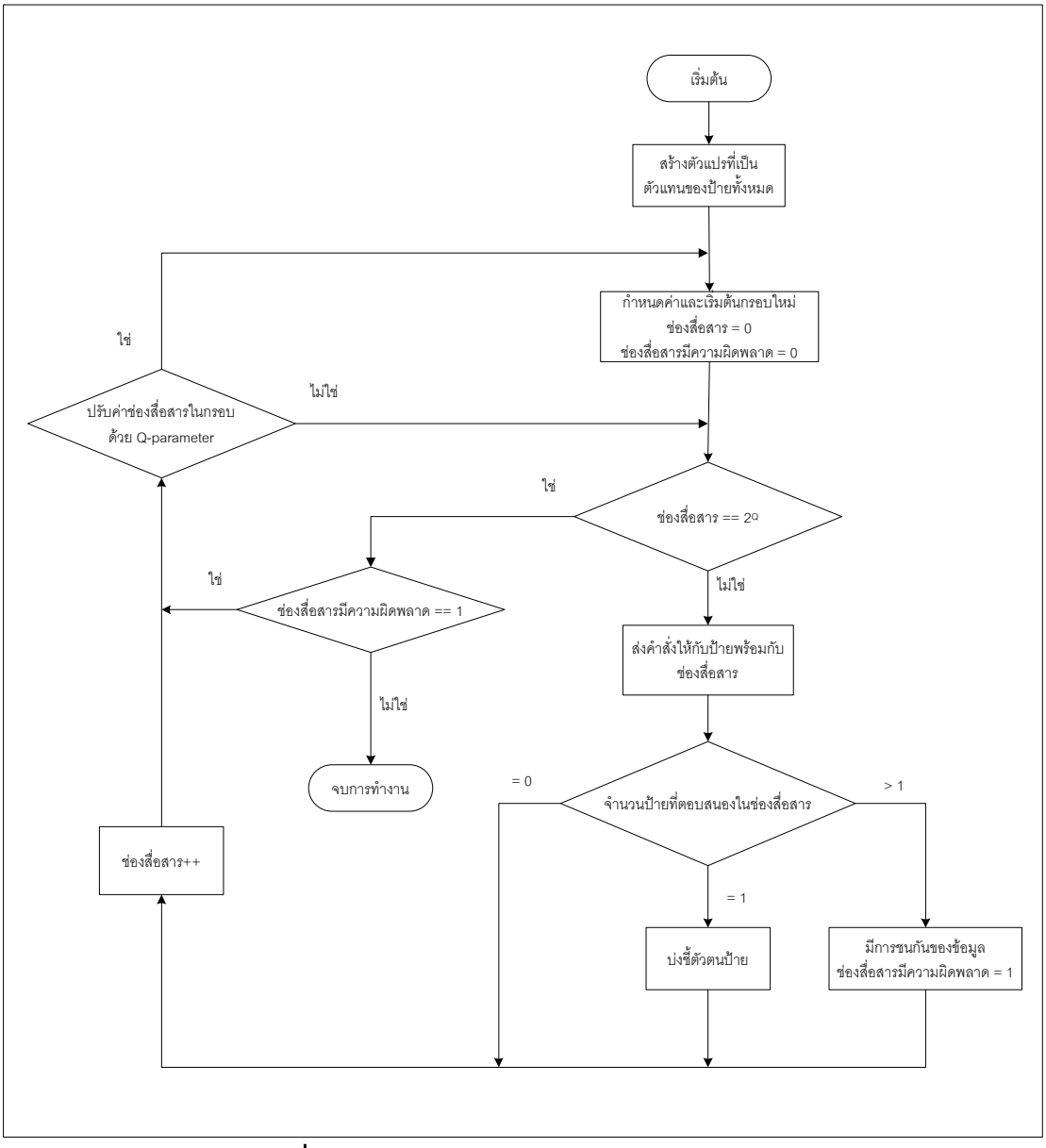
2. มีป้ายตอบสนอง 1 ป้าย

โปรแกรมจะทำการบ่งชี้ตัวตนป้ายที่ตอบสนองและทำการส่งคำสั่งพร้อมทั้งช่องสื่อสารถัดไป

3. มีป้ายตอบสนองมากกว่า 1 ป้าย

โปรแกรมจะทำส่งคำสั่งพร้อมทั้งช่องสื่อสารถัดไปให้กับป้าย

โปรแกรมจะทำงานตามเหตุการณ์ด้านบนไปจนกว่า จะครบตามช่องสื่อสารที่กำหนดไว้ จากนั้นจึงตรวจสอบถ้ามีช่องสื่อสารใดในกรอบมีการชนกันของข้อมูลก็จะทำการเริ่มการทำงานใหม่โดยการเริ่มกรอบใหม่ และทำการเก็บจำนวนคำสั่งที่ใช้งานในการป้องกันการชนกันของข้อมูลเพื่อมาเปรียบเทียบกับการป้องกันการชนกันของข้อมูลแบบอื่นๆ แต่ถ้าในกรอบไม่มีการชนกันของข้อมูลก็จะเสร็จสิ้นการทำงานของโปรแกรม




**รูปที่** 3.8 แผนผังการจำลองการทำงานของ DFSA

### 3.2.4 อัลกอริทึม Dynamic Framed Slotted Aloha

ในการจำลองการทำงานของการป้องกันการชนกันของข้อมูลแบบ DFSA นั้นเนื่องจาก DFSA ได้มีการนำไปใช้งานในมาตรฐาน 18000-6 ประเภท C หรือ EPC Class 1 Generation 2 ดังนั้นผู้วิจัยจึงได้ทำการจำลองการทำงานตามที่ระบุไว้ในมาตรฐาน เริ่มต้นโดยสร้างตัวแปรเพื่อเก็บค่าตัวแปรที่สำคัญของป้ายทั้งหมดในระยะการทำงานได้แก่ จำนวนช่องสื่อสารที่สุ่มได้ 16 บิต, และ พารามิเตอร์ Q จากนั้นจึงเริ่มกระบวนการป้องกันการชนกันของข้อมูล โดยส่งคำสั่งให้กับป้ายพร้อมกับค่าของช่องสื่อสารที่ใช้อยู่ถ้าโดยค่าของช่องสื่อสารที่ใช้อยู่จะมีค่าตั่งแต่ 0 ถึง $2^Q - 1$ ช่องสื่อสาร โดยเริ่มต้นจากช่องสื่อสารที่ 0 เป็นลำดับแรก เครื่องอ่านจะทำการส่งคำสั่งให้ป้ายทำการสุ่มตัวเลขจำนวน 16 บิตซึ่งเป็นตัวแทนของช่องสื่อสารที่ป้ายจะทำการส่งข้อมูล ถ้าป้ายใดมีช่องสื่อสารที่สุ่มได้เท่ากับช่องสื่อสารที่โปรแกรมส่งไปก็จะทำการตอบสนองกับมาดังรูปที่ 3.8 ซึ่งเหตุการที่อาจเกิดขึ้นได้ มี 3 กรณีดังต่อไปนี้

1. ไม่มีป้ายตอบสนอง

   โปรแกรมจะทำส่งคำสั่งพร้อมทั้งช่องสื่อสารถัดไปให้กับป้าย ถ้าป้ายใดมีช่องสื่อสารที่สุ่มได้เท่ากับช่องสื่อสารที่โปรแกรมส่งไปก็จะทำการตอบสนองกับเครื่องอ่าน

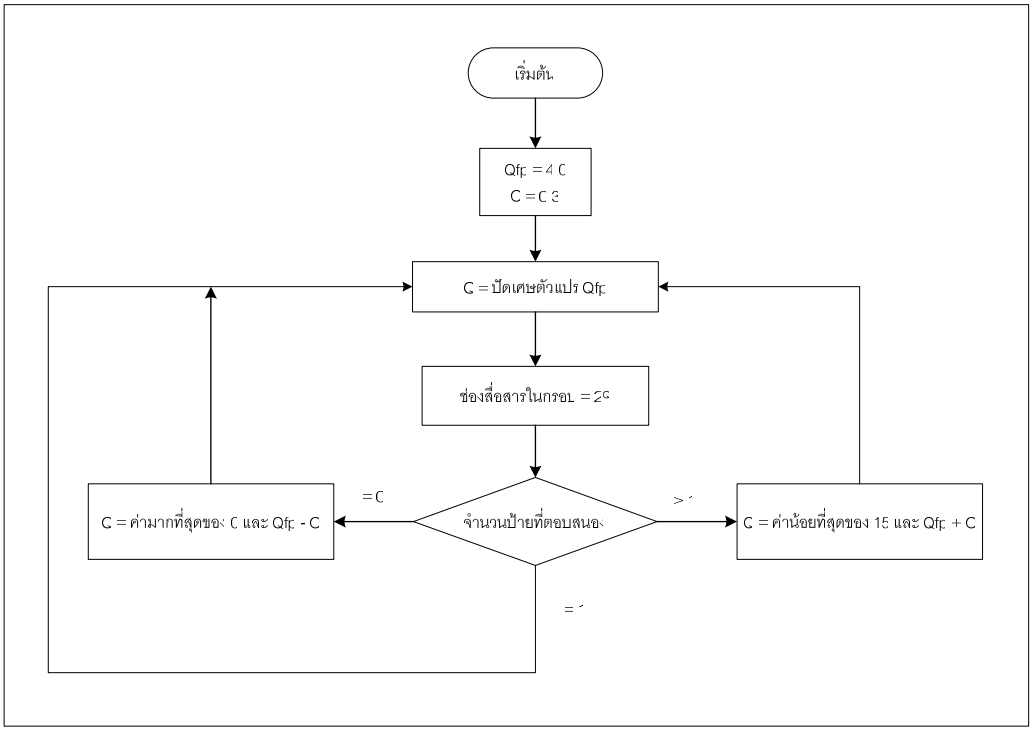
2. มีป้ายตอบสนอง 1 ป้าย

   โปรแกรมจะทำการบ่งชี้ตัวตนป้ายที่ตอบสนองและทำการส่งคำสั่งพร้อมทั้งค่าของช่องสื่อสารถัดไป ถ้าป้ายใดมีช่องสื่อสารที่สุ่มได้เท่ากับช่องสื่อสารที่โปรแกรมส่งไปก็จะทำการตอบสนองกับเครื่องอ่าน

3. มีป้ายตอบสนองมากกว่า 1 ป้าย

   โปรแกรมจะทำส่งคำสั่งพร้อมทั้งช่องสื่อสารถัดไปให้กับป้าย ถ้าป้ายใดมีช่องสื่อสารที่สุ่มได้เท่ากับช่องสื่อสารที่โปรแกรมไปป้ายจะตอบสนองกับเครื่องอ่าน

โปรแกรมจะทำงานตามเหตุการณ์ที่กำหนดไว้ จากนั้นจะเข้าสู่การปรับเปลี่ยนขนาดของช่องสื่อสารในกรอบโดยใช้ พารามิเตอร์ Q ในการทำงาน ซึ่งจะเป็นไปตามรูปที่ 3.7 ขนาดของช่องสื่อสารในกรอบจะมีจำนวนเท่ากับ $2^Q$ เมื่อมีการชนกันของข้อมูลหลายครั้งก็จะทำการปรับค่าพารามิเตอร์ Q เพิ่มขึ้น แต่ถ้าไม่มีป้ายตอบสนองกลับมาก็ทำการลดค่า พารามิเตอร์ Q ลง โดยจะ



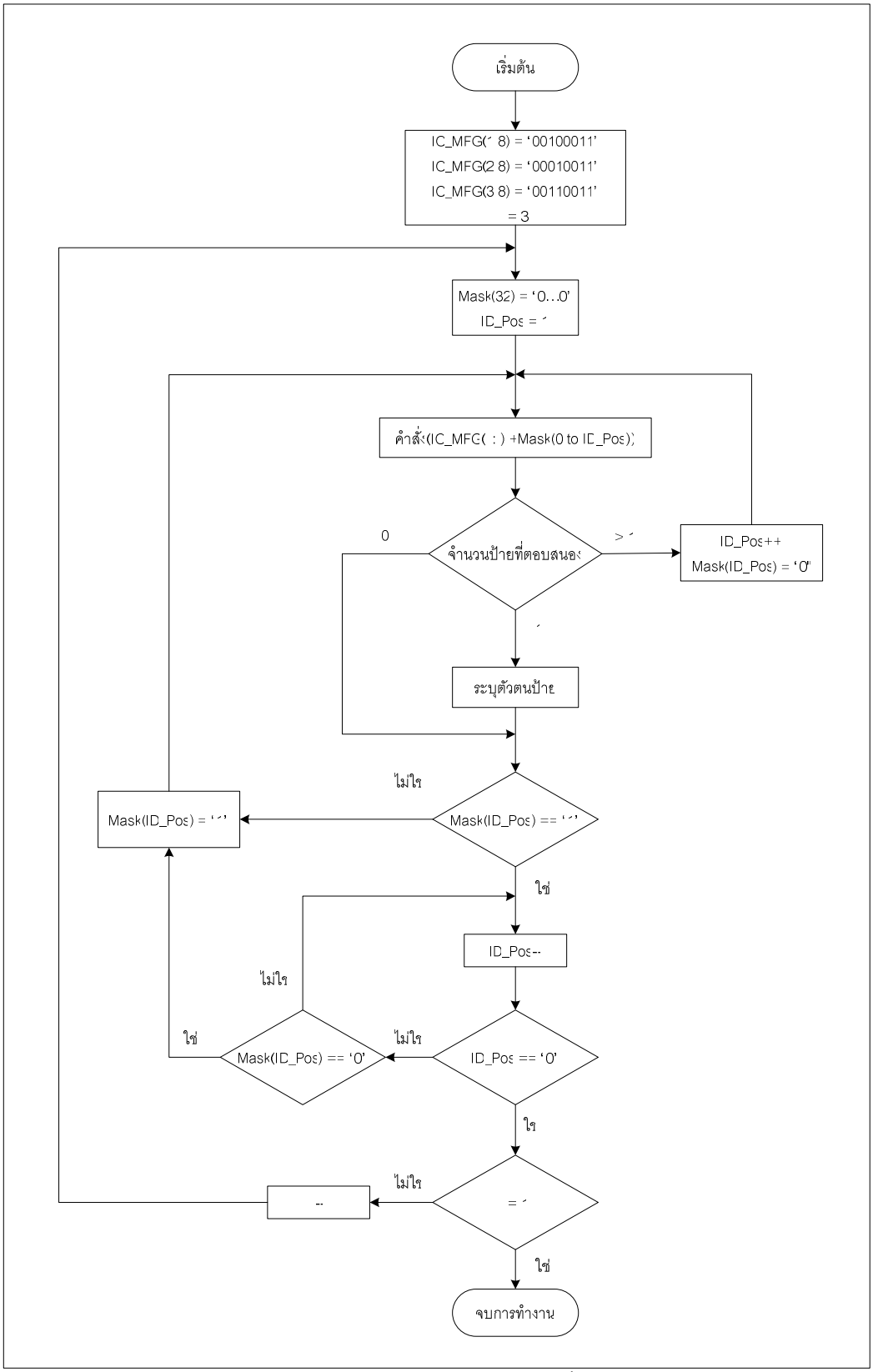

**รูปที่** 3.7 แผนผังการจำลองการทำงานของพารามิเตอร์ Q

ใช้ค่า C ในการเพิ่มหรือลดค่าของ พารามิเตอร์ Q ซึ่งค่าคงที่ C จะอยู่ในช่วงระหว่าง 0.1 ถึง 0.5 ในการจำลองการทำงานผู้วิจัยได้เลือกใช้ค่าคงที่ C เท่ากับ 0.3 โดยรายละเอียดนั้นได้แสดงไว้ในบทที่ 2 จากนั้นโปรแกรมจะทำการป้องกันการชนกันของข้อมูลจนครบตามที่กำหนดไว้ จากนั้นจึงตรวจสอบถ้ามีช่องสื่อสารใดในกรอบมีการชนกันของข้อมูลก็จะทำการเริ่มการทำงานใหม่โดยการเริ่มกรอบใหม่ และทำการเก็บจำนวนคำสั่งที่ใช้งานในการป้องกันการชนกันของข้อมูล เพื่อมาเปรียบเทียบกับการป้องกันการชนกันของข้อมูลแบบอื่นๆ แต่ถ้าในกรอบไม่มีการชนกันของข้อมูลก็จะเสร็จสิ้นการทำงานของโปรแกรม

## 3.3 การพัฒนาการทำงานของอัลกอริทึมป้องกันการชนกันของข้อมูล

### 3.3.1 อัลกอริทึมแบบใหม่ที่ใช้ข่าวสารล่วงหน้า (A New Anti-Collision Based on A-Priori Information)

ในการทำงานของ อัลกอริทึมต้นไม้แบบทวิภาคที่กล่าวไปในบทที่ 2 นั้นจะใช้การค้นหาไปตามรหัสประจำตัวของป้าย ซึ่งมีลักษณะเป็นเลขฐานสอง ซึ่งถ้าเราทราบข้อมูลบางส่วนของรหัส




**รูปที่** 3.9 แผนผังการทำงานของอัลกอริทึมแบบใหม่ที่ใช้ข่าวสารล่วงหน้า [15]

ประจำตัวของป้ายเราก็จะลดระยะเวลาการทำงานของ BT ได้ ดังนั้นผู้ทำการวิจัยจึงได้นำข้อดีนี้มาพัฒนาการทำงานของ อัลกอริทึมต้นไม้แบบทวิภาคให้มีการทำงานที่รวดเร็วมากยิ่งขึ้น [15] ในรูปที่ 2.13 ได้แสดงส่วนประกอบของรหัสประจำตัวป้าย ของมาตรฐาน ISO 18000-6 ประเภท A และ B จากรูปจะส่วนที่เราจะทราบนั้นแป็นส่วนของ IC Mfg code ซึ่งเป็นตัวเลขของบริษัทผู้ผลิตดังนั้นข้อมูลส่วนนี้เราจะสามารถทราบล่วงหน้าได้ ในการทำงานของโปรแกรมตามรูปที่ 3.9 นั้นเราจะต้องทราบข้อมูลบริษัทผู้ผลิตจำนวน 8 บิต ซึ่งเก็บไว้ในตัวแปร IC_MFG ซึ่งในโปรแกรมตัวอย่างมี 3 ชุด เมื่อเริ่มต้นโปรแกรมจะต้องกำหนดค่าตัวแปร Mask และ ID_Pos เป็น “00…0” และ 1 ตามลำดับ โดยตัวแปร Mask จะเป็นตัวแทนของ รหัสประจำตัวป้ายจำนวน 32 บิต ส่วน ID_Pos นั้นเป็นตัวแทนของจำนวนบิตที่ต้องการนำไปเปรียบเทียบ เช่น เมื่อต้องการเปรียบเทียบเฉพาะรหัสประจำตัวป้าย 3 ตัวหน้าต่อจากรหัสบริษัท ID_Pos ก็จะเท่ากับ 3 เมื่อกำหนดค่าตัวแปรเรียบร้อยแล้ว ก็จะเริ่มต้นกระบวนการป้องกันการชนกันของข้อมูล โดยจะส่งรหัสประจำตัวป้ายซึ่งประกอบไปด้วย 2 ส่วนคือ IC_MFG และ Mask(0 ถึง ID_Pos) เมื่อป้ายได้รับคำสั่งดังกล่าวจึงจะทำการเปรียบเทียบรหัสประจำตัวป้ายที่ได้รับ กับ รหัสประจำตัวป้ายของป้ายเองถ้าตรงกันก็จะตอบสนองกลับมา โดยจะตอบสนองเป็นรหัสประจำตัวป้ายของตัวเองทั้งหมด 64 บิต จากนั้นเครื่องอ่านสามารถวิเคราะห์จากสัญญาณได้ว่ามีป้ายตอบสนอง 1 ป้าย, มากกว่า 1 ป้าย หรือ ไม่มีป้ายตอบสนอง จากนั้นจึงตัดสินใจทำตามขั้นตอนดังต่อไปนี้

1. ไม่มีป้ายตอบสนอง (Idle)

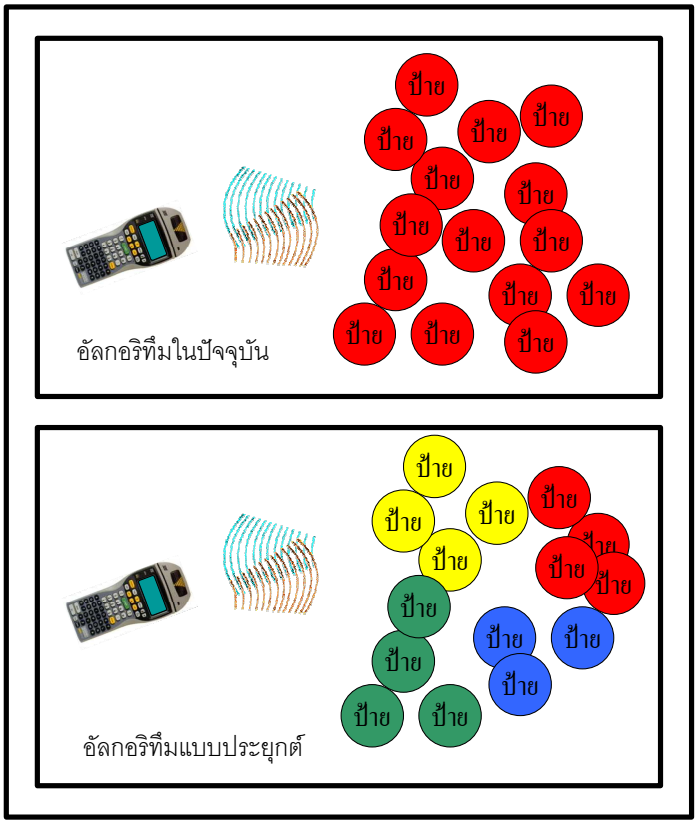
   โปรแกรมจะทำการเปลี่ยนบิต Mask ที่ตำแหน่ง ID_Pos จาก “0” เป็น “1” แต่ถ้า เป็น “1” อยู่แล้วจะทำการลดตำแหน่งของ ID_Pos ไปที่ละ 1 จนกว่า Mask ที่ตำแหน่ง ID_Pos จะเท่ากับ “0” แล้วจึงเปลี่ยนเป็น “1” แต่ถ้าลดค่า ID_Pos จนเป็น 0 ก็จะออกจากโปรแกรม

2. มีป้ายตอบสนอง 1 ป้าย (Success)

   เมื่อมีป้ายตอบสนอง 1 ป้ายโปรแกรมจะทำการบ่งชี้ตัวตนป้ายนั้นโดย เก็บรหัสประจำตัวป้ายไว้ในฐานข้อมูล จากนั้นจึงเปลี่ยนบิตเหมือนกับขั้นตอนที่ไม่มีป้ายตอบสนอง

3. มีป้ายตอบสนองมากกว่า 1 ป้าย (Fail)

   เมื่อมีป้ายตอบสนองมากกว่า 1 ป้าย โปรแกรมจะทำการเพิ่ม ID_Pos ขึ้นไปอีก 1 จากนั้นจึงกำหนดค่า Mask ที่ตำแหน่ง ID_Pos เป็น “0”

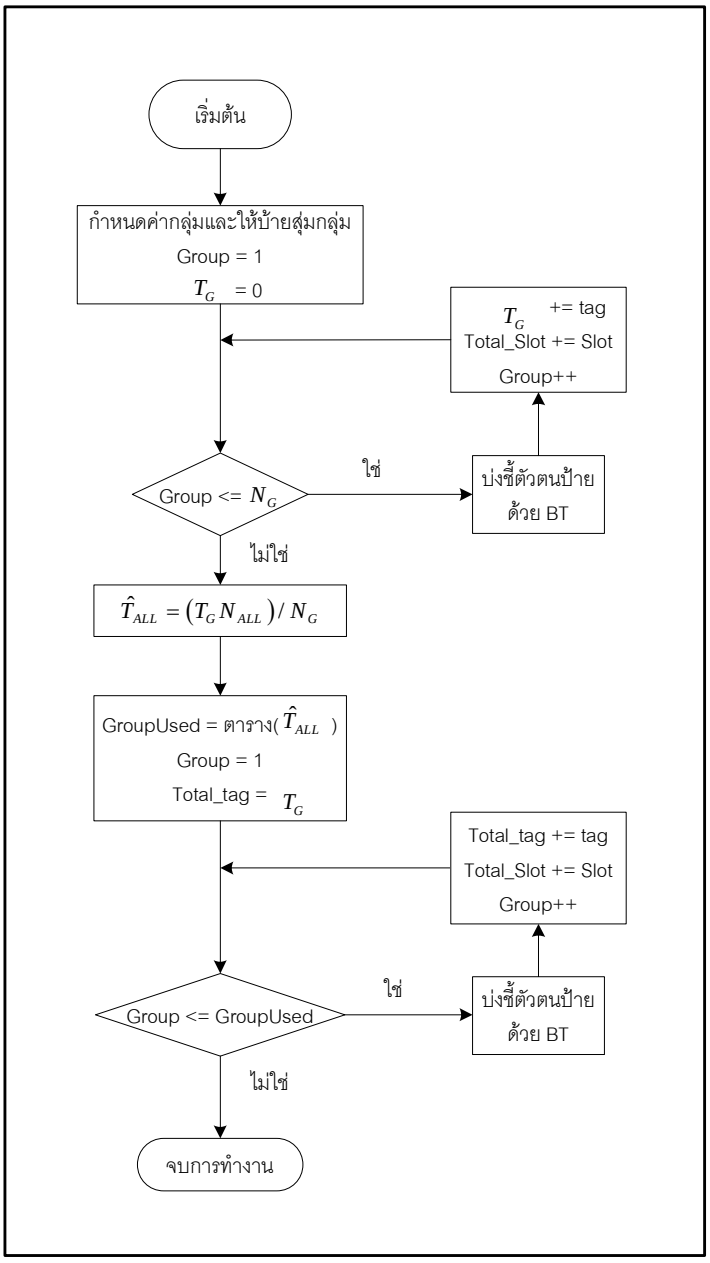



**รูปที่** 3.10 แผนผังการทำงานของโปรแกรม [16]

เมื่อทำการปรับเปลี่ยนค่าตัวแปร Mask เรียบร้อยแล้วโปรแกรม จะทำการส่งคำสั่งพร้อมกับตัวแปร Mask ให้กับป้ายเพื่อรอผลตอบสนอง และจะทำตามขั้นตอนนี้จนสามารถบ่งชี้ตัวตนป้ายได้ครบทุกป้าย

### 3.3.2 อัลกอริทึมแบบประยุกต์ที่ใช้กับจำนวนป้ายอาร์เอฟไอดีหนาแน่นมาก (A Novel Anti – collision Algorithm for High - Density RFID Tags)

ในการทำงานของ BT และ DFSA นั้นมีทั้งข้อดีและข้อเสียในการพัฒนาอัลกอริทึมแบบใหม่นี้ผู้วิจัยจะทำการนำข้อดีของทั้งสองอัลกอริทึมมาใช้งานโดยข้อดีของ BTนั้นจะทำงานได้ดีในกรณีที่มีป้ายจำนวนน้อยและหมายเลขรหัสประจำตัวป้ายแตกต่างกัน ส่วนอัลกอริทึม DFSA นั้นจะทำงานได้โดยไม่สนใจหมายเลขรหัสประจำตัวป้ายทำให้ทำงานได้ดี ในกรณีที่หมายเลขรหัสประจำตัวป้ายส่วนหน้าซ้ำกัน ดังนั้นผู้วิจัยจึงนำวิธีการทำงานของอัลกอริทึมทั้งสองมาทำงานรวมกันซึ่งเรียกว่าอัลกอริทึมแบบประยุกต์ โดยทำการนำป้ายทั้งหมดที่อยู่ในระยะการอ่านของ




**รูปที่** 3.11 แผนผังการทำงานของโปรแกรม [16]

เครื่องอ่านมาทำการแบ่งกลุ่มโดยอาศัยการทำงานแบบ DFSA จากนั้นจึงใช้อัลกอริทึมแบบ BT มาทำการบ่งชี้ตัวตนป้ายของแต่ละกลุ่มดังรูปที่ 3.10 [16] ในกระบวนการทำงานของอัลกอริทึมนั้นได้แสดงไว้ดังรูปที่ 3.11 เมื่อเริ่มต้นโปรแกรมจะทำการกำหนดค่ากลุ่มเริ่มต้นเพื่อทำการหาค่าประมาณของจำนวนป้ายที่อยู่ในระยะการทำงานของเครื่องอ่าน โดยกำหนดกลุ่มเริ่มต้นที่ 32 กลุ่มจากนั้นจึงใช้สมการที่ 3.1 หาค่าประมาณของจำนวนป้าย

$$\hat{T}_{ALL} = (T_G N_{ALL}) / N_G \quad (3.1)$$

โดยที่ $\hat{T}_{ALL}$ คือค่าประมาณของจำนวนป้ายทั้งหมดในระยะการอ่านของเครื่องอ่าน

$T_G$ คือจำนวนป้ายที่บ่งชี้ตัวตนได้จากการบ่งชี้ตัวตน $N_G$ กลุ่ม

$N_G$ คือจำนวนกลุ่มที่ทำการบ่งชี้ตัวตน ($N_G$ = 3)

$N_{ALL}$ คือจำนวนกลุ่มทั้งหมดที่ทำการบ่งชี้ตัวตน ($N_{ALL}$ = 32)

เมื่อทำการได้ค่าประมาณจำนวนป้ายทั้งหมดในระยะการอ่านของเครื่องอ่านแล้วจะทำการเปิดตารางเพื่อหากลุ่มที่เหมาะสมในการทำงานโดยกลุ่มที่เหมาะสมจะแสดงในบทที่ 4 จากนั้นเครื่องอ่านจะทำการส่งค่ากลุ่มให้กับป้าย ป้ายจะทำการสุ่มค่ากลุ่มของแต่ละป้ายเองจากนั้นเครื่องอ่านจะเริ่มกระบวนการทำงานอัลกอริทึม BT ในแต่ละกลุ่มจากนั้นจึงเก็บจำนวนช่องสื่อสารเพื่อมาเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึม

## 3.4 การสร้างอัลกอริทึมป้องกันการชนกันของข้อมูลในเครื่องอ่าน

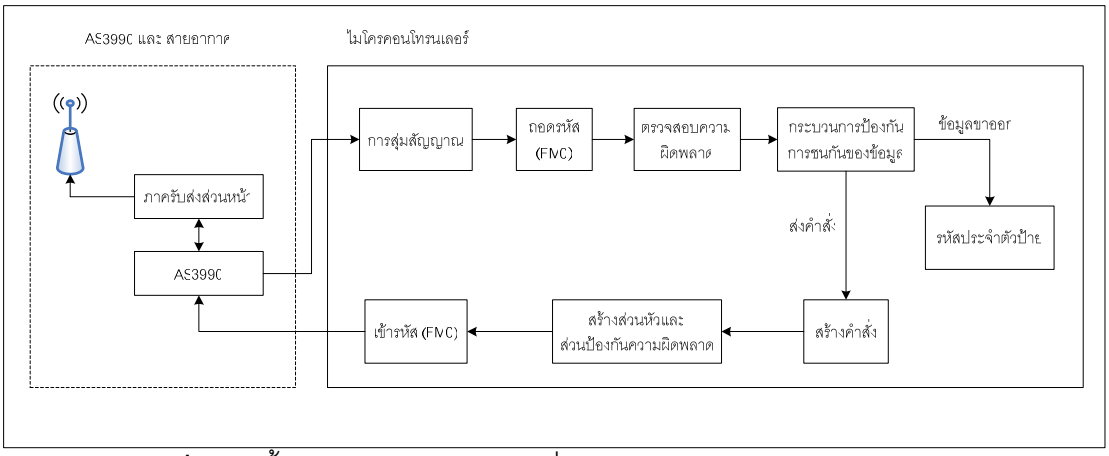
การสร้างอัลกอริทึมป้องกันการชนกันของข้อมูลในเครื่องอ่านนั้น ผู้ทำการวิจัยได้ทำการสร้างอัลกอริทึมป้องกันการชนกันของข้อมูลตามมาตรฐานที่ใช้กันซึ่งมีอยู่ 3 มาตรฐาน ดังต่อไปนี้

### 3.4.1 มาตรฐาน ISO 18000-6 ประเภท A

ในปัจจุบันการใช้งานของป้ายประเภทนี้นั้นได้เลิกใช้งานไปแล้ว ดังนั้นในงานวิจัยนี้จะไม่ทำการสร้างอัลกอริทึมป้องกันการชนกันของข้อมูลในเครื่องอ่านในมาตรฐานนี้

### 3.4.2 มาตรฐาน ISO 18000-6 ประเภท B

ในการสร้างอัลกอริทึมป้องกันการชนกันของข้อมูลในมาตรฐาน ISO 18000-6 ประเภท B นั้น ผู้ทำการวิจัยได้ทำการสร้างอัลกอริทึมโดยใช้ภาษาซี ที่ใช้กับไมโครคอนโทรลเลอร์ ตระกูล MSP 430 ซึ่งมีขั้นตอนการทำงานของเครื่องอ่านดังรูปที่ 3.12




**รูปที่** 3.12 ขั้นตอนการทำงานของเครื่องอ่านในมาตรฐาน ISO 18000-6 B

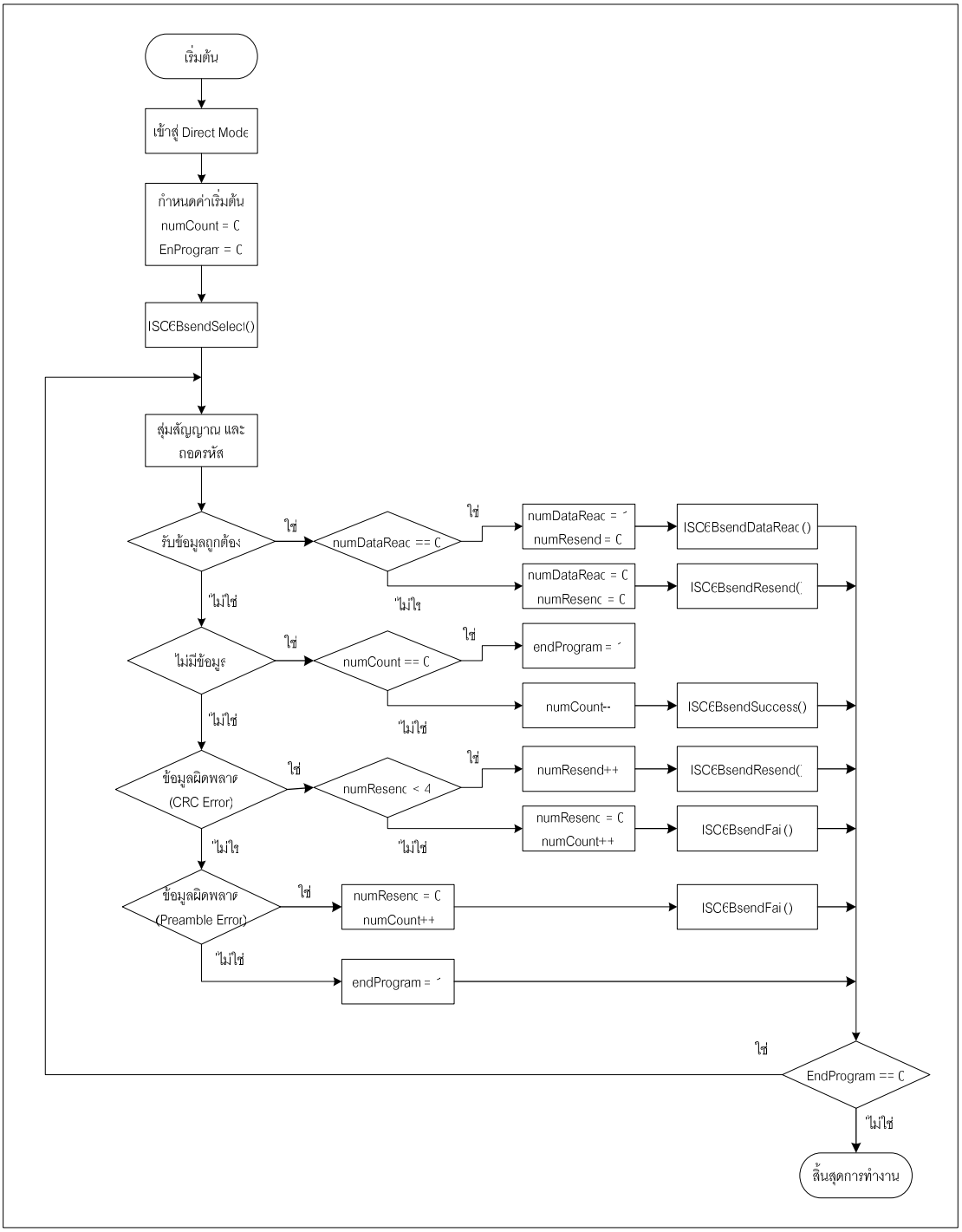
จากรูปที่ 3.12 ได้แสดงขั้นตอนการทำงานของเครื่องอ่านป้าย ตามมาตรฐาน ISO 18000-6 ประเภท B โดยจะแบ่งการทำงานออกเป็นสองส่วนคือ

1. การทำงานของ AS3990 และ สายอากาศ
2. การทำงานของไมโครคอนโทรลเลอร์

เมื่อเริ่มต้นการทำงานภาครับของเครื่องอ่านจะทำการส่งคำสั่งให้ป้ายในระยะการทำงานของเครื่องอ่าน โดยในการส่งคำสั่งนั้นจะต้องชุดคำสั่งขึ้น จากนั้นจึงทำการสร้างส่วนหัว และ ส่วนป้องกันความผิดพลาด ให้กับชุดคำสั่ง จากนั้นจึงทำการเข้ารหัสแบบ FM0 ตามมาตรฐานที่มีความเร็วในการรับส่งข้อมูลที่ 40 Kbit/s จากนั้นจึงส่งข้อมูลให้กับ AS3990 เพื่อทำการเปลี่ยนสัญญาณดิจิตอลเป็นอนาล็อก แล้วส่งออกผ่านสายอากาศให้กับป้าย

ส่วนการทำงานในภาครับของเครื่องอ่านนั้นเมื่อได้รับของสัญญาณอนาล็อก ชิป AS3990 จะทำการเปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล จากนั้นจึงส่งให้กับไมโครคอนโทรลเลอร์ทำการสุ่มสัญญาณ และถอดรหัสข้อมูล (FM0) จากนั้นจึงตรวจสอบความผิดพลาดของข้อมูลที่ได้รับมา เมื่อได้รับข้อมูลเรียบร้อยแล้วจะทำการส่งข้อมูลเข้าสู่กระบวนการป้องกันการชนกันการชนกันของข้อมูลเพื่อตัดสินใจการทำงานขั้นต่อไปของเครื่องอ่าน ซึ่งถ้าไม่มีการชนกันของข้อมูลเครื่องอ่านจะสามารถบ่งชี้ตัวตนของป้ายได้

ในการทำงานของโปรแกรมนั้นจะเริ่มจากไมโครคอลโทรลเลอร์ทำการกำหนดค่าตัวแปรเพื่อให้ชิป AS3990 ทำงานใน Direct โหมด จากนั้นจึงกำหนดค่าตัวแปรต่างๆ และเริ่มต้นการป้องกันการชนกันของข้อมูลโดยจะเริ่มจากการเลือกป้ายที่อยู่ในระยะอ่านเข้ามาทำการป้องกันการชนกันของข้อมูลด้วยคำสั่ง ISO6BsendSelect() โดยในการใช้คำสั่งนี้จะเลือกเฉพาะป้ายที่มี รหัสประจำตัวตรงกับรหัสประจำตัวที่อยู่ในคำสั่งเข้ามากทำการป้องกันการชนกันของข้อมูล แต่ถ้าส่ง




**รูปที่** 3.13 แผนผังการทำงานของโปรแกรมป้องกันการชนกันของข้อมูล
ตามมาตรฐาน ISO 18000-6 ประเภท B

รหัสประจำตัวป้ายเป็นเลขฐานสิบหกที่เท่ากับ 0x0000000000000000 จะเป็นการเลือกป้ายทั้งหมดที่อยู่ในระยะการทำงานเข้ามาทำการป้องกันการชนกันของข้อมูล เมื่อส่งคำสั่งนี้แล้วป้ายทั้งหมดที่อยู่ในระยะการทำงานของเครื่องอ่านจะทำการส่งรหัสประจำตัวพร้อมกับรหัสตรวจสอบความผิดพลาด เมื่อเครื่องอ่านได้รับสัญญาณชิป AS3990 จะทำการแปลงสัญญาณอนาล็อคเป็นสัญญาณดิจิตอล จากนั้นไมโครคอลโทรลเลอร์จะทำการสุ่มสัญญาณที่ได้รับและ ถอดรหัสโดยมีส่วนที่สำคัญ 3 ส่วนที่จะต้องทำการตรวจสอบ คือ 1. มีสัญญาณตอบกลับมาหรือไม่ 2. สามารถตรวจสอบรหัสส่วนหน้า (Preamble) ได้หรือไม่ 3. ตรวจสอบความผิดพลาดได้ถูกต้องหรือไม่ จากนั้นจึงนำมาวิเคราะห์เพื่อทำตามขั้นตอนต่อไปนี้

1. สามารถรับข้อมูลได้ถูกต้อง

- ตัวแปร numDataRead มีค่าเท่ากับ 0

  โปรแกรมจะทำการเก็บรหัสป้ายนั้นๆจากนั้นจะส่งคำสั่ง ISO6BsendDataRead พร้อมกับ รหัสป้ายนั้นๆ เพื่อเปลี่ยนสถานะป้ายไปที่ Data Exchange และกำหนดค่า numDataRead = 1 และ numResend = 0

- ตัวแปร numDataRead มีค่าไม่เท่ากับ 0

  โปรแกรมจะทำการระบ่งชี้ป้ายนั้นๆจากนั้นจะส่งคำสั่ง ISO6BsendResend เพื่อตรวจสอบในกลุ่มยังมีป้ายเหลืออีกหรือไม่ และกำหนดค่า numDataRead = 0 และ numResend = 0

2. ไม่มีป้ายตอบกลับ

- ตัวแปร numCount มีค่าเท่ากับ 0

  จะสิ้นสุดการทำงานของโปรแกรมเนื่องจาก ไม่มีป้ายเหลือในกลุ่มสุดท้ายที่ต้องบ่งชี้ตัวตน

- ตัวแปร numCount มีค่าไม่เท่ากับ 0

  จะทำการส่งคำสั่ง ISO6BsendSuccess เพื่อทำการลดค่าตัวแปรนับในป้ายลง 1 ค่า หรืออีกนัยหนึ่งก็คือเลือกกลุ่มต่อไปมาทำการระบุตัวตน

3. มีความผิดพลาดของข้อมูล

- ตัวแปร numResend มีค่าน้อยกว่า 4

  โปรแกรมจะส่งคำสั่ง ISO6BsendResend พร้อมกับลดค่าตัวเพิ่มค่าตัวแปร numResend อีก 1 ค่า

- ตัวแปร numDataRead มีค่ามากกว่าหรือเท่ากับ 4

  โปรแกรมจะส่งคำสั่ง ISO6BsendFail เพื่อเพิ่มค่าตัวแปรนับ ในป้ายหรืออีกนัยหนึ่งคือการให้กลุ่มที่ทำงานอยู่แบ่งกลุ่มออกเป็น 2 กลุ่ม และ เพิ่มค่าตัวแปร numCount อีก 1 ค่าและกำหนดค่า numResend = 0

4. มีความผิดพลาดที่ส่วนหน้าของชุดข้อมูล

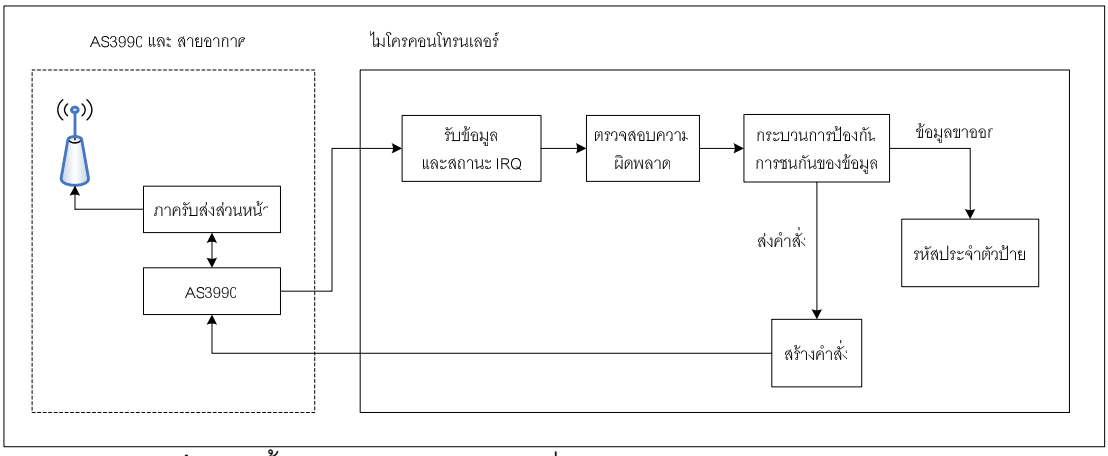
   โปรแกรมจะส่งคำสั่ง ISO6BsendFail เพื่อเพิ่มค่าตัวแปรนับในป้ายหรืออีกนัยหนึ่งคือการให้กลุ่มที่ทำงานอยู่แบ่งกลุ่มออกเป็น 2 กลุ่ม และเพิ่มค่าตัวแปร numCount อีก 1 ค่าและกำหนดค่า numResend = 0

โปรแกรมจะทำตามขั้นตอน 1 – 4 ไปจนสามารถบ่งชี้ตัวตนของป้ายในกลุ่มของป้ายที่เลือก ได้ทั้งหมดจึงจบการทำงานของโปรแกรม

### 3.4.3 มาตรฐาน ISO 18000-6 ประเภท C

ในการสร้างอัลกอริทึมป้องกันการชนกันของข้อมูลในมาตรฐาน ISO 18000-6 ประเภท C นั้น ผู้ทำการวิจัยได้ทำการสร้างอัลกอริทึมโดยใช้ภาษาซี ที่ใช้กับไมโครคอนโทรลเลอร์ ตระกูล MSP 430 ซึ่งมีขั้นตอนการทำงานของเครื่องอ่านดังรูปที่ 3.14

ในการทำงานของโปรแกรมนั้นจะเริ่มจากไมโครคอลโทรลเลอร์ทำการกำหนดค่าตัวแปรเพื่อให้ชิป AS3990 ทำงานใน Normal โหมด จากนั้นจึงกำหนดค่าตัวแปรต่างๆ และเริ่มต้นการป้องกันการชนกันของข้อมูลโดยจะเริ่มจากการเลือกป้ายที่อยู่ในระยะอ่านเข้ามาทำการป้องกันการชนกันของข้อมูลด้วยคำสั่ง Select โดยในการใช้คำสั่งนี้จะเลือกเฉพาะป้ายที่มี รหัสประจำตัวตรงกับรหัสประจำตัวที่อยู่ในคำสั่งเข้ามากทำการป้องกันการชนกันของข้อมูล เมื่อส่งคำสั่งนี้แล้วป้ายทั้งหมดที่อยู่ในระยะการทำงานของเครื่องอ่าน จากนั้นเครื่องอ่านจะส่งคำสั่ง Query พร้อมกับพารามิเตอร์ Q เพื่อเริ่มต้นกระบวนการป้องกันการชนกันของข้อมูลแบบ DFSA จากนั้นป้ายจะทำการสุ่มช่องสื่อสาร และส่งรหัสที่ทำการสุ่มขึ้นเพื่อเป็นตัวแทนของป้ายพร้อมกับรหัสตรวจสอบความผิดพลาดเมื่อตรงช่องสื่อสารของป้ายมีค่าเท่ากับ 0 จากนั้นเครื่องอ่านจะได้รับสัญญาณที่ตอบสนองจากป้าย ชิป AS3990 จะทำการแปลงสัญญาณอนาล็อคเป็นสัญญาณดิจิตอล, ทำการถอดรหัสข้อมูลตามาตรฐาน ISO 18000-6 ประเภท C และ ถอดส่วนหัวของข้อมูลออก จากนั้นจึงส่งข้อมูลให้กับไมโครคอลโทรลเลอร์ทำการ ตรวจสอบความผิดพลาดของข้อมูล จากนั้นจึงนำมา




**รูปที่** 3.14 ขั้นตอนการทำงานของเครื่องอ่านในมาตรฐาน ISO 18000-6 C

วิเคราะห์เพื่อทำตามสถานะของข้อมูลที่ได้รับจากชิป AS3990 ดังรูปที่ 3.15 โดยผ่านตัวแปรแสดงสถานะของตัวแปร IRQ

1. ไม่มีป้ายตอบสนองในช่องสื่อสาร

   เครื่องอ่านจะทำการส่งคำสั่ง QueryRep ให้กับป้าย

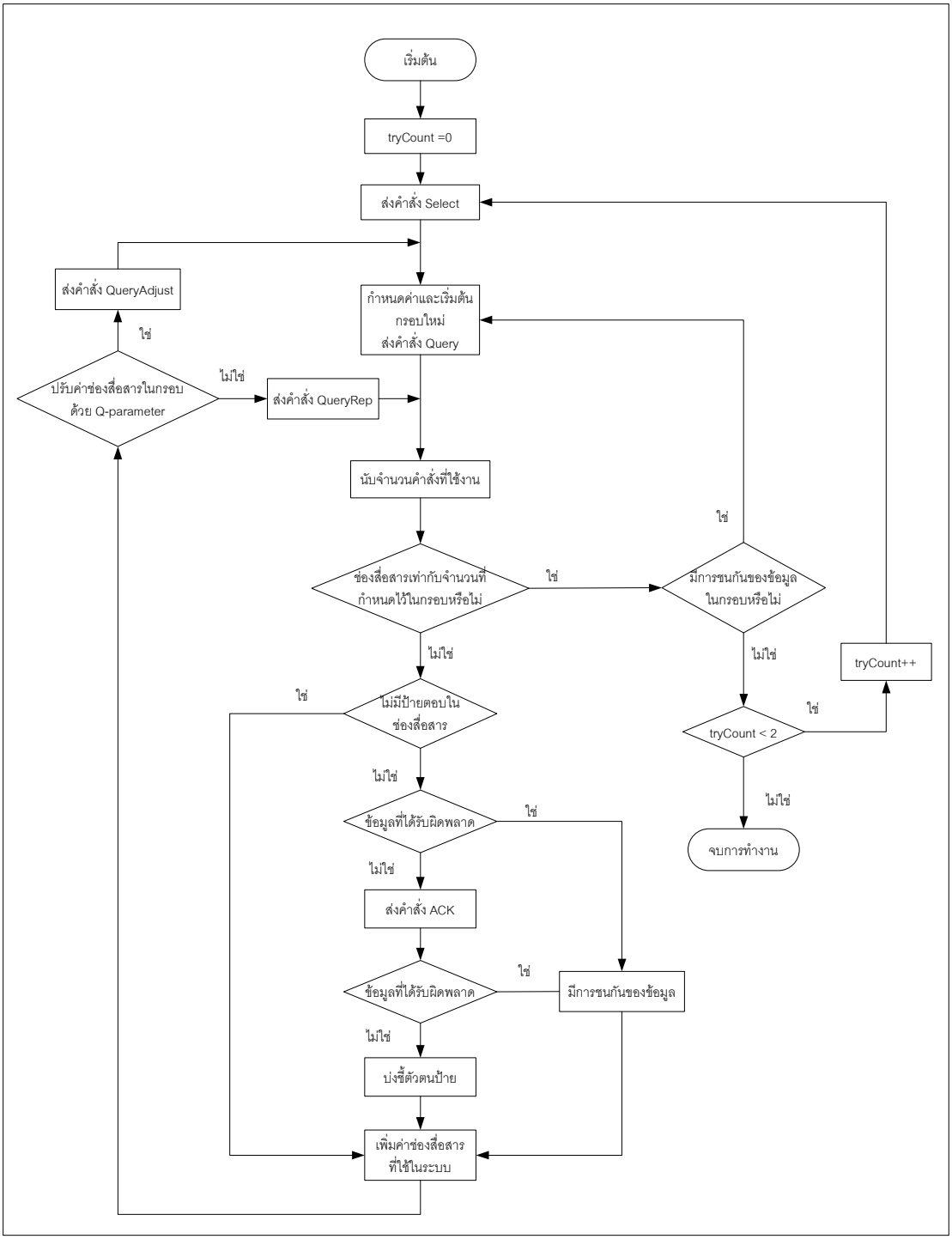
2. ได้รับข้อมูลผิดพลาด

   เครื่องอ่านจะทำการส่งคำสั่ง QueryRep ให้กับป้าย

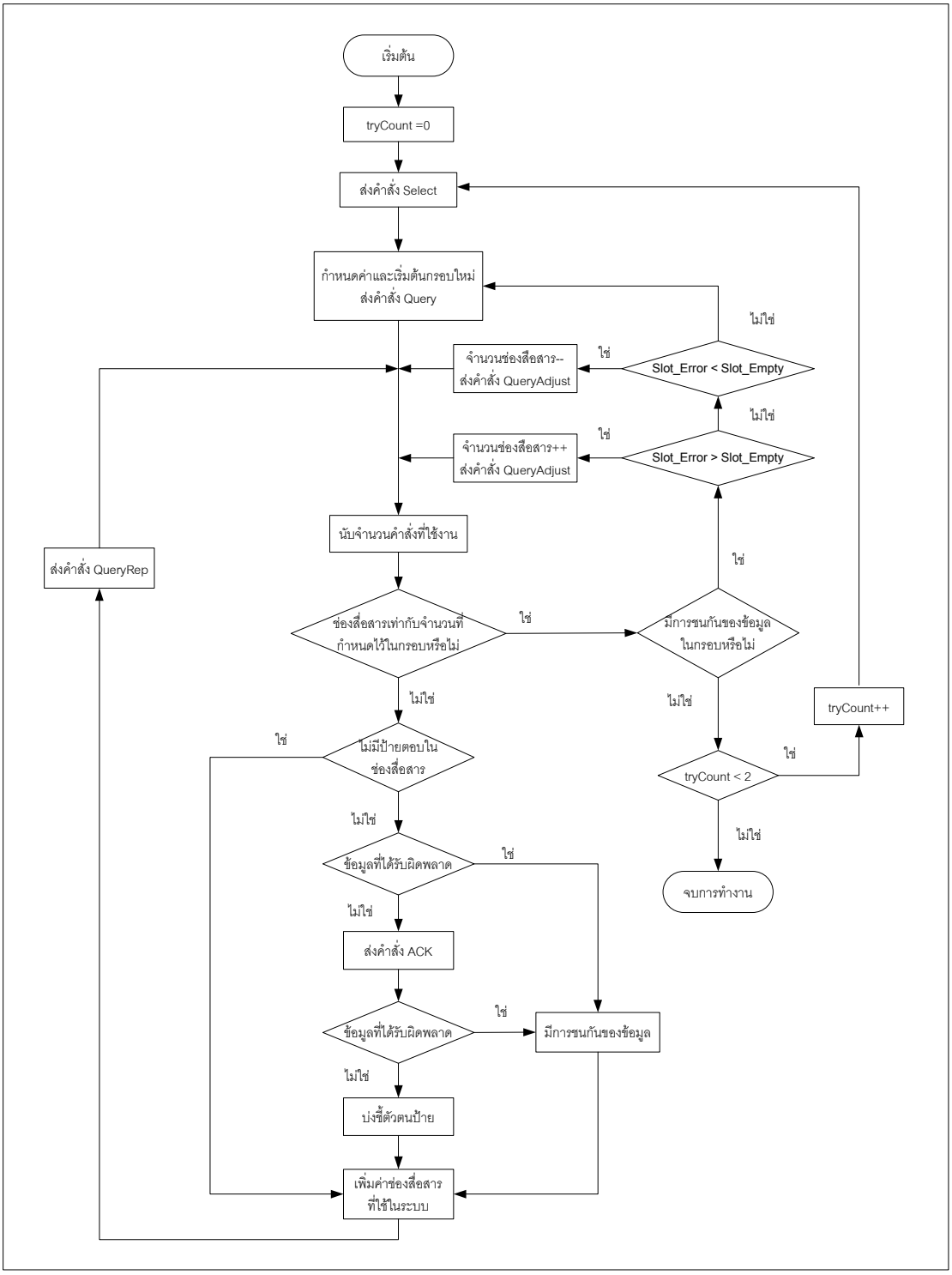
3. ได้รับข้อมูลถูกต้อง

   เครื่องอ่านจะทำการส่งคำสั่ง ACK พร้อมกับรหัสที่ป้ายทำการสุ่มขึ้นเพื่อเป็นตัวแทนของป้ายจากนั้นป้ายจะทำการตอบสนองพร้อมกับรหัสประจำตัวป้ายจำนวน 96 บิต จากนั้นเครื่องอ่านจะส่งคำสั่ง QueryRep ให้กับป้าย

จากนั้นเครื่องอ่านจะทำการปรับเปลี่ยนจำนวนช่องสื่อสารโดยอาศัย พารามิเตอร์ Q โดยขนาดของช่องสื่อสารในกรอบจะมีจำนวนเท่ากับ $2^{\text{พารามิเตอร์ Q}}$ เมื่อมีการชนกันของข้อมูลหลายครั้งก็จะทำการปรับค่าพารามิเตอร์ Q เพิ่มขึ้น แต่ถ้าไม่มีป้ายตอบสนองกลับมาก็ทำการลดค่าพารามิเตอร์ Q ลง โดยจะใช้ค่า C ในการเพิ่มหรือลดค่าของ พารามิเตอร์ Q ซึ่งค่า C จะอยู่ในช่วงระหว่าง 0.1 ถึง 0.5 โดยในโปรแกรมที่ใช้งานเลือกใช้ค่า C เท่ากับ 0.1 จากนั้นเครื่องอ่านจะทำการบ่งชี้ตัวตนป้าย อีกหนึ่งครั้งเพื่อให้มีความถูกต้องมากยิ่งขึ้น ผู้วิจัยได้ทำการออกแบบอัลกอริทึมที่ใช้งานในเครื่องอ่านในมาตรฐาน ISO 18000 – 6 ประเภท C 2 แบบ คือ อัลกอริทึมมาตรฐาน จะทำงานตรงตามมาตรฐาน ISO 18000 – 6 ประเภท C ตามที่กล่าวไป และ อัลกอริทึมแบบวิเคราะห์ผลการชนกันของข้อมูล นั้นจะเป็นไปดังรูปที่ 3.16 โดยจะทำงานแต่ต่างจากอัลกอริทึมมาตรฐาน คือ จะไม่ใช้พารามิเตอร์ Q ในการปรับเปลี่ยนจำนวนช่องสื่อสารในกรอบ แต่จะใช้ข้อมูล




**รูปที่** 3.15 แผนผังการทำงานของโปรแกรมป้องกันการชนกันของข้อมูลตามมาตรฐาน ISO 18000-6 ประเภท C หรือ อัลกอริทึมมาตรฐาน




**รูปที่** 3.16 แผนผังการทำงานของอัลกอริทึมป้องกันการชนกันของข้อมูล
แบบวิเคราะห์ผลการชนกันของข้อมูล

จำนวนช่องสื่อสารที่ว่างและจำนวนช่องสื่อสารที่มีการชนกันของข้อมูลมาทำการเปรียบเทียบเมื่อจบการทำงานของทุกช่องสื่อสารในกรอบเรียบร้อยแล้ว เมื่อจำนวนช่องสื่อสารที่ว่างมากกว่าจำนวนช่องสื่อสารที่มีการชนกันของข้อมูลจะทำการลดจำนวนช่องสื่อสารในกรอบ แต่ถ้าจำนวนช่องสื่อสารที่ว่างเปล่าน้อยกว่าจำนวนช่องสื่อสารที่ชนกันของข้อมูลจะทำการเพิ่มจำนวนช่องสื่อสารในกรอบ

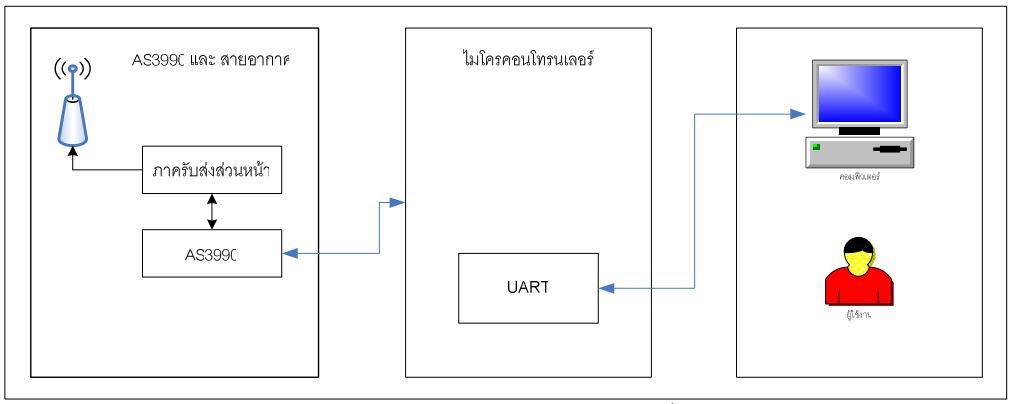
## 3.4 โปรแกรมควบคุมการทำงานของเครื่องอ่าน

ในการควบคุมการทำงานของเครื่องอ่านนั้นผู้วิจัยได้ทำการเขียนโปรแกรมวิชวลเบสิกเพื่อทำการควบคุมการทำงานของเครื่องอ่านของบริษัท Austriamicrosystems รูปที่ 3.17 แสดงแผนผังการทำงานของเครื่องอ่าน โดยชิป AS3990 นั้นจะถูกควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ ขั้นตอนแรกไมโครคอนโทรลเลอร์จะทำการกำหนดค่าเริ่มต้นให้กับชิป AS3990 จากนั้นจะรอคำสั่งการทำงานจากผู้ใช้งานผ่านคอมพิวเตอร์ โดยใช้การสื่อสารโดยใช้พอร์ตอนุกรมที่ความเร็ว 115200 บิตต่อวินาที

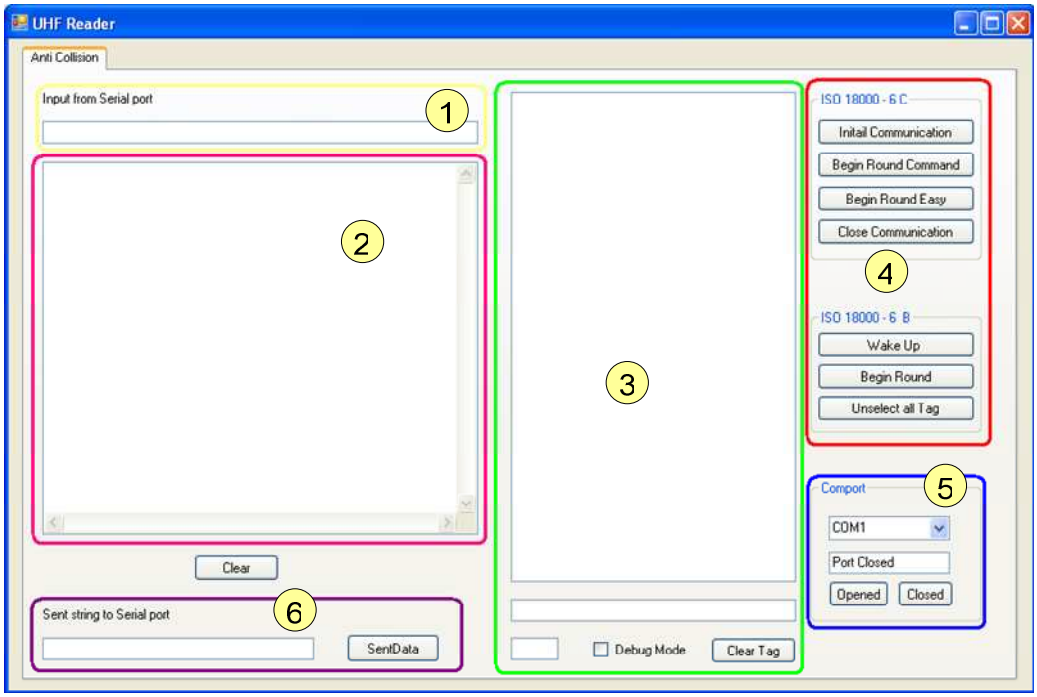
ในส่วนติดต่อผู้ใช้งานของโปรแกรมควบคุมการทำงานของเครื่องอ่านนั้นได้มีการออกแบบไว้ดังรูปที่ 3.18 โดยจะแบ่งออกเป็น 3 ส่วนคือ

1. ส่วนกำหนดค่าเริ่มต้นและติดต่อกับเครื่องอ่านผ่านพอร์ตอนุกรม

   ในส่วนนี้จะทำการกำหนดค่าเริ่มต้นและทำการติดต่อกับเครื่องอ่านผ่านพอร์ตอนุกรมที่มีความเร็วเท่ากับ 115200 บิตต่อวินาที ดังรูปที่ 3.18 ในส่วนที่ 5




**รูปที่** 3.17 แผนผังการทำงานของเครื่องอ่าน




**รูปที่** 3.18 ส่วนติดต่อผู้ใช้งานของโปรแกรมควบคุมเครื่องอ่าน

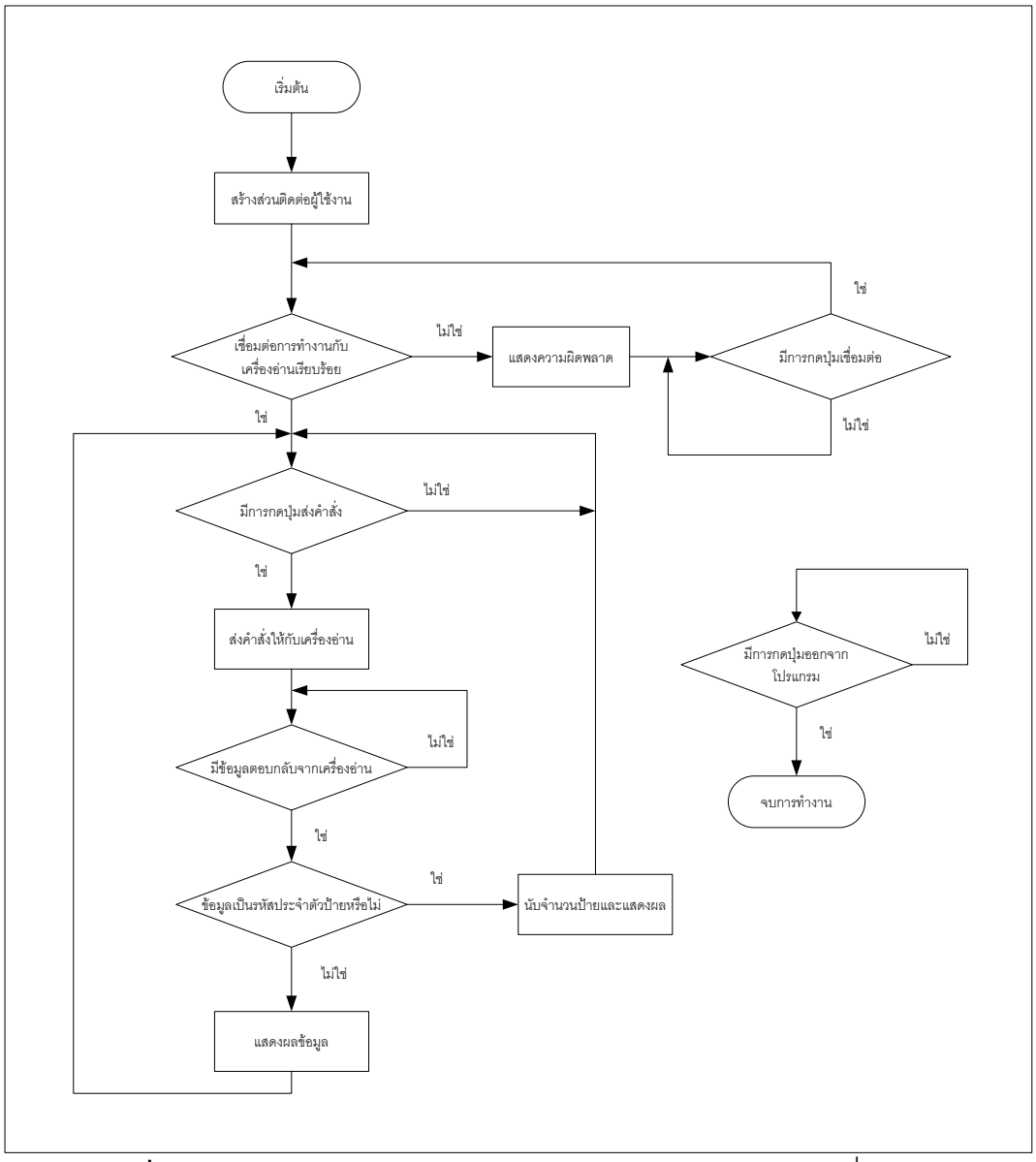
2. ส่วนส่งคำสั่งให้กับเครื่องอ่าน

ส่วนส่งคำสั่งให้กับเครื่องอ่านนั้นแบ่งออกเป็น 2 ส่วนคือ ส่วนที่ 4 และ 6 ดังรูปที่ 3.18 ในส่วนที่ 4 นั้นเป็นปุ่มควบคุมให้เครื่องอ่าน อ่านป้ายโดยใช้อัลกอริทึมที่ได้เขียนไว้ดังที่กล่าวไว้ในหัวข้อ 3.3 การสร้างอัลกอริทึมป้องกันการชนกันของข้อมูลในเครื่องอ่าน ส่วนที่ 6 นั้นเป็นการส่งคำสั่งให้กับเครื่องอ่านโดยตรง

3. ส่วนแสดงผลข้อมูลที่ได้รับจากเครื่องอ่าน

ส่วนแสดงผลข้อมูลแบ่งออกเป็น 2 ส่วนคือ ส่วนที่ส่งข้อมูลออกจากโปรแกรมผ่านพอร์ตอนุกรม ไปให้กับเครื่องอ่านส่วนนี้จะแสดงในรูปที่ 3.18 ในส่วนที่ 2 โดยในส่วนที่ 2 นี้จะแสดงข้อมูลที่ได้รับจากเครื่องอ่านด้วยตามลำดับก่อนหลังของการส่งข้อมูล ในรูปที่ 3.18 ส่วนที่ 1 จะเป็นส่วนแสดงผลข้อมูลที่ได้รับจากเครื่องอ่าน ส่วนสุดท้ายส่วนที่ 3 ในรูปที่ 3.18 ทำหน้าที่แสดงรหัสประจำตัวป้ายแต่ละป้ายและบอกถึงจำนวนป้ายที่บ่งชี้ตัวตนได้

ส่วนของโครงสร้างการทำงานของโปรแกรมควบคุมเครื่องอ่านนั้นได้แสดงไว้ดังรูปที่ 3.19 เมื่อเริ่มโปรแกรมแล้ว โปรแกรมจะทำการสร้างส่วนติดต่อผู้ใช้งานตามรูปที่ 3.18 จากนั้นจึงทำการติดต่อกับเครื่องอ่านผ่านพอร์ตอนุกรม ถ้าไม่สามารถติดต่อเครื่องอ่านได้จะแสดงความ




**รูปที่** 3.19 แผนผังการทำงานของโปรแกรมควบคุมการทำงานของเครื่องอ่าน

ผิดพลาด จากนั้นจึงรอผู้ใช้งานทำการเชื่อมต่อ เมื่อเชื่อมต่อได้แล้ว โปรแกรมจะหยุดรอให้ผู้ใช้งานทำการส่งคำสั่งการทำงานในรูปที่ 3.18 ส่วนที่ 4 และ 6 เมื่อผู้ใช้งานทำการส่งคำสั่งออกไปแล้ว โปรแกรมจะทำการรอเก็บข้อมูลที่เครื่องอ่านตอบกลับมา ถ้าคำสั่งที่ส่งไปเป็นคำสั่งอ่านรหัสประจำตัวป้าย เมื่อเครื่องอ่านส่งรหัสประจำตัวป้ายให้กับโปรแกรมแล้ว โปรแกรมจะทำการนับจำนวนป้ายและแสดงผลรหัสประจำตัวป้ายในส่วนที่ 3 ตามรูปที่ 3.18 เมื่อผู้ใช้งานต้องการออกจากโปรแกรมควรทำการปิดการทำงานของพอร์ตอนุกรมในส่วนที่ 5 ตามรูปที่ 3.18 และปิดการทำงานของโปรแกรม เป็นการสิ้นสุดการทำงานของโปรแกรม