

การจัดการระบบฐานข้อมูลด้วยบาร์โค้ดและรถขนส่งอัตโนมัติ
(Database Managment By Barcode and Automatic Transport)

นางสาวฐิติรัตน์	คุ้มสุพรรณ
นายเทียนชัย	กวมรัมย์
นายนาวิน	โคมฉาย
นายปริญญา	สุวรรณคันธ์
นายภัทรพล	ศรีกัลยาณบุตร

อาจารย์ที่ปรึกษา อาจารย์สมภพ รอดอัมพร

โครงการวิศวกรรมศาสตร์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตร์บัณฑิต
สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

พ.ศ. 2543

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ปัจจุบัน โปรแกรมจัดการฐานข้อมูลได้เข้ามามีบทบาทในการจัดเก็บข้อมูลที่มีจำนวนมากซึ่งในการจัดเก็บข้อมูลด้วย คอมพิวเตอร์มีข้อดีมากกว่า เมื่อเทียบกับการจัดเก็บข้อมูลด้วยเอกสารซึ่งก็สามารถจัดเก็บที่เป็นระเบียบทำให้การค้นหาข้อมูลและการแก้ไขข้อมูลก็สามารถทำได้ด้วยความรวดเร็ว โอกาสในการผิดพลาดและการสูญหายของข้อมูลก็น้อยลง ซึ่งจะเห็นได้ว่าการจัดเก็บข้อมูลที่มีจำนวนมากนั้นมีความสำคัญมาก ดังนั้นการจัดเก็บข้อมูลที่ดีและมีประสิทธิภาพ จึงควรเก็บด้วยคอมพิวเตอร์

ในการเก็บข้อมูลแผนกเวชระเบียนก็ต้องอาศัยบุคลากรจำนวนมากเข้ามาดำเนินงาน ในการจัดเก็บข้อมูลซึ่งก็คือการเพิ่มข้อมูล การแก้ไขข้อมูล รวมไปถึงการค้นหาข้อมูลจากเอกสารซึ่งมีจำนวนมากขึ้นทุกวันทำให้มีการทำงานที่ล่าช้า และยากแก่การแก้ไข การค้นหาและการเก็บรักษาเอกสารซึ่งถ้าเอกสารดังกล่าวหมดอายุ ก็ทำให้ยากแก่การทำลายและก่อให้เกิดขยะและมลภาวะที่เกิดจากการเผาอีกด้วยระบบฐานข้อมูล เป็นแนวทางการแก้ไขปัญหาดังกล่าวในการจัดเก็บข้อมูลซึ่งช่วยลดค่าใช้จ่ายเกี่ยวกับเอกสาร ลดเวลาการแก้ไขและการค้นหาข้อมูล อีกทั้งลดจำนวนบุคลากรในการทำงานลง จึงเป็นผลทำให้การผิดพลาดและการสูญหายของข้อมูลลดน้อยลงด้วย การเพิ่มข้อมูลการแก้ไขข้อมูล การลบข้อมูลและการแสดงผลของข้อมูลทางเครื่องพิมพ์ เป็นต้น ในการจัดเก็บข้อมูลจะเก็บเป็น 2 หัวข้อ คือ การจัดเก็บประวัติคนไข้ ข้อมูลการเข้าพักรักษาในโรงพยาบาล และจัดลำดับคิวที่จะทำการส่งผู้ป่วยไปยังแต่ละแผนกได้ ทำให้มีการทำงานที่มีประสิทธิภาพมากยิ่งขึ้น และยังช่วยลดมลภาวะที่จะเกิดขึ้นอีกด้วย

ในขณะเดียวกันนี้ระบบการขนส่งทะเบียนประวัติผู้ป่วยไปยังแผนกต่างๆ ก็เป็นขั้นตอนการดำเนินงานที่สำคัญขั้นตอนหนึ่ง ซึ่งระบบขนส่งทะเบียนประวัติผู้ป่วยของโรงพยาบาลส่วนใหญ่ยังใช้แรงงานคน จึงอาจกล่าวได้ว่าเป็นการสิ้นเปลืองทั้งทรัพยากรบุคคลและงบประมาณ

ดังนั้นการเลือกระบบขนส่งที่มีเครื่องจักรกลเข้ามาช่วยในการขนส่งทะเบียนประวัติผู้ป่วยไปยังแผนกต่างๆ ระบบขนส่งที่ใช้ในปัจจุบันจึงควรจะเป็นรถขนส่งที่มีขนาดเหมาะสมกับงาน สะดวกในการขนถ่าย มีน้ำหนักไม่มากจนเกินไปและสามารถขนส่งตามเส้นทางที่กำหนด โดยปราศจากแรงงานคนมาควบคุม จึงพอจะสรุปแนวทางได้ว่า การขนส่งดังกล่าวมาแล้วน่าจะมียานขนส่งอัตโนมัติซึ่งจะมีคุณสมบัติตามที่ได้อธิบายมาแล้ว

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อสร้างโปรแกรมจัดการฐานข้อมูลแผนกเวชระเบียนซึ่งจะจัดเก็บประวัติของคนไข้ที่เข้ามาทั้งหมดได้

1.2.2 เพื่อให้สามารถค้นหาข้อมูลคนไข้ได้รวดเร็วด้วยระบบบาร์โค้ด

1.2.3 เพื่อความเป็นระเบียบในการจัดลำดับที่จะส่งคนไข้ไปในแผนกต่างๆ

1.2.4 เพื่อการสร้างรถขนส่งอัตโนมัติ ที่ช่วยในการขนส่งทะเบียนประวัติผู้ป่วยไปยังแผนกต่างๆ

1.2.5 เพื่อช่วยลดบุคลากรในการจัดการระบบฐานข้อมูลแผนกเวชระเบียน และบุคลากรในการลงทะเบียนประวัติคนไข้ไปยังแผนกต่างๆ

1.2.6 เพื่อสร้างรถขนส่งอัตโนมัติต้นแบบและนำไปพัฒนาต่อไป

1.3 ขอบเขตของโครงการ

1.3.1 ขอบเขตของระบบฐานข้อมูลแผนกเวชระเบียน

ระบบฐานข้อมูลแผนกเวชระเบียนนี้จะทำงานเกี่ยวกับการจัดเก็บข้อมูลประวัติของคนไข้ที่เข้ามาทำการรักษา และการจัดระบบคิวเพื่อส่งไปยังแผนกต่างๆ เพื่อให้เกิดความสะดวกรวดเร็ว และถูกต้องในการทำงาน โดยจะประกอบด้วย

1.3.1.1 การเก็บประวัติของคนไข้ที่เข้ามาทำการรักษา โดยป้อนรายละเอียด ผ่านบนฟอร์มที่ได้ออกแบบไว้ทางหน้าจอ

1.3.1.2 สามารถตรวจเช็คประวัติของคนไข้ และข้อมูลการเข้าพักในศูนย์การแพทย์ โดยระบุวันเวลาที่เข้าพักได้ด้วยระบบบาร์โค้ด หรือพิมพ์ชื่อ

1.3.1.3 จัดลำดับคิว โดยพิมพ์เป็นบัตรออกมาระบุหมายเลขประจำตัวผู้ป่วย ชื่อ แผนก และลำดับคิวได้

1.3.2 ขอบเขตของรถขนส่งทะเบียนประวัติ

รถขนส่งทะเบียนประวัติผู้ป่วยภายในโรงพยาบาลที่ใช้สวิทช์หรือคีย์บอร์ดในการทำงานให้รถวิ่งไปตามเส้นทางที่กำหนดไว้ ด้วยการรับคำสั่งควบคุมทิศทางการวิ่งจากวงจรตรวจจับเส้นทางแถบสีด้วยแสงอินฟราเรด และวงจรหยุดรถเมื่อถึงสถานีที่กำหนดไว้ สัญญาณที่ได้รับจะถูกประมวลผลโดย ไมโครคอนโทรลเลอร์ MCS-51 แล้วส่งออกไปให้ส่งให้วงจรควบคุมมอเตอร์ทำงาน โดยรถขนส่งนั้นจะบรรทุกน้ำหนักได้ไม่เกิน 10 กิโลกรัม

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 เพิ่มความรวดเร็ว และความปลอดภัยในการจัดเก็บรายชื่อ และประวัติของคนไข้

1.4.2 เพื่อปรับปรุงประสิทธิภาพการให้บริการ ด้านทะเบียนประวัติ และการทำงานด้าน
เวชระเบียน

1.4.3 ฐานข้อมูลที่สามารถจัดเก็บประวัติคนไข้ได้เพียงพอต่อไปในอนาคต

1.4.4 ช่วยลดบุคลากรในการควบคุม การจัดเก็บข้อมูล และขนส่งทะเบียนประวัติผู้ป่วยภายใน
โรงพยาบาล

1.4.5 เพิ่มความรวดเร็วและสะดวกให้กับคนไข้ที่มาทำการรักษาและญาติคนไข้

1.4.6 ทำให้เกิดความสะดวกในการขนส่งทะเบียนประวัติผู้ป่วยภายในโรงพยาบาล

1.4.7 รถขนส่งอัตโนมัติต้นแบบ

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ฮาร์ดแวร์ (Hardware)

ในระบบฐานข้อมูลที่มีประสิทธิภาพควรมีฮาร์ดแวร์ต่างๆที่พร้อมจะอำนวยความสะดวกในระบบงานและงานข้อมูลได้อย่างมีประสิทธิภาพ ไม่ว่าจะเป็นขนาดของหน่วยความจำหลัก ความเร็วของหน่วยประมวลผลกลาง (Central Processing Unit : CPU) หน่วยความจำสำรองที่จะรองรับการประมวลผลของข้อมูล และอีกอย่างหนึ่งที่ขาดเสียไม่ได้คือ อุปกรณ์อินพุตและเอาต์พุต ซึ่งในโครงการนี้มีอุปกรณ์อินพุต คือ คีย์บอร์ดและหัวอ่านแถบรหัสบาร์โค้ดรวมไปถึงแถบรหัสบาร์โค้ดแบบ 3 ใน 9 ด้วยรายละเอียดของอุปกรณ์ฮาร์ดแวร์ต่างๆมีดังนี้

2.1.1 เครื่องอ่านรหัสแถบบาร์โค้ด

องค์ประกอบทั่วไปที่สำคัญของเครื่องอ่านรหัสแถบได้แก่

ก) อิเล็กตรออปติกส์ (Elector - Optics) คือเลนส์ที่รับแสงสะท้อนจากแถบ และช่องว่างของรหัสแถบ ไปเป็นสัญญาณทางไฟฟ้า ภายนอกจากนั้นระบบนี้ยังรวมไปถึงแหล่งที่มาของแสง ตัวแปลงเป็นสัญญาณและระบบนี้ยังรวมไปถึงแหล่งที่มาของแสง ตัวแปลงเป็นสัญญาณและระบบเลนส์รับแสงอีกชุดหนึ่ง ที่ใช้เพื่อรวมแสงให้เป็นจุดเดียวกัน

ข) ส่วนปรับเปลี่ยนสัญญาณ (Signal Conditioning) คือระบบการปรับเปลี่ยนสัญญาณจากค่าความกว้างของแถบและช่องว่างเปลี่ยนไปเป็นสัญญาณจากอะนาลอก (Analog) และเปลี่ยนเป็นสัญญาณดิจิทัล (Digital) อีกที่ซึ่งกระบวนการนี้เรียกว่า การปรับรูปคลื่น (Wave Shaping)

ค) ส่วนประมวลผลข้อมูล หลังจากเครื่องอ่านข้อมูลถอดรหัส แล้วข้อมูลก็สามารถเก็บไว้ในเครื่องอ่านข้อมูลเพื่อรอที่จะส่งผ่านเข้าคอมพิวเตอร์ในภายหลังได้หรือจะต่อเครื่องอ่านข้อมูลโดยตรงไปยังคอมพิวเตอร์เลยก็ได้

2.1.1.1 ชนิดของเครื่องอ่านรหัสแถบ

เครื่องอ่านรหัสแถบโดยทั่วไปแบ่งออกเป็น 4 ชนิดคือ

ก) เครื่องอ่านข้อมูลระบบแสงคงที่แบบมือถือ (Hand-held fixed beam)

ข) เครื่องอ่านข้อมูลระบบแสงคงที่แบบติดตั้ง (Stationary Fixed Beam)

- ค) เครื่องอ่านข้อมูลระบบแสงเคลื่อนที่แบบ ติดตั้ง (Stationary Moving Beam)
 ง) เครื่องอ่านข้อมูลระบบแสงเคลื่อนที่แบบมือถือ (Hand-Held Moving Beam)



รูปที่ 2.1 เครื่องอ่านข้อมูลระบบแสงคงที่แบบมือถือ (Hand-Held Fixed Beam)

แสงเลเซอร์แบบเฮลิอัมเนออน (Helium Neon Laser : HNL) หรืออินฟราเรดไดโอด (Infrared Diode) ใช้พลังงานต่ำที่สิ้นกระเพื่อม ผ่านจอภาพไปยังรหัสเพื่ออ่านข้อมูล โกวินที่ด้ามมีรูปร่างเป็นจะ ทำหน้าที่ บังคับให้เครื่องอ่านทำงาน แม้ว่าเครื่องอ่านข้อมูลระบบแสงเคลื่อนที่แบบมือถือชนิดปืน เลเซอร์นี้จะมีราคาแพงกว่าและซับซ้อนมากกว่าเครื่องอ่านข้อมูลระบบคงที่ แต่ก็มีข้อดีในด้านการ นำ ไปใช้งานคือ สามารถอ่านรหัสได้ในระยะไกลถึง 25 นิ้ว นอกจากนั้นลำแสงที่เคลื่อนที่จะกวาด อ่านรหัสได้หลายๆครั้งต่อวินาที และบางครั้งก็สามารถอ่านกับรหัสที่มีคุณภาพในการพิมพ์ฉลากต่ำ ได้ด้วย ดังรูปที่ 2.2 เครื่องอ่านข้อมูลชนิดปืนเลเซอร์นี้สามารถจะนำไปติดตั้งไว้กับเครื่องอ่านข้อมูลแบบติดตั้งอยู่ก็ได้แต่ปืนเลเซอร์ที่นำติดตัวได้จะใช้งานได้ สะดวกรวดเร็วกว่า และข้อมูลที่เก็บรวบรวมได้ก็สามารถจะส่งเข้า เครื่องคอมพิวเตอร์ได้หลังการเก็บข้อมูล

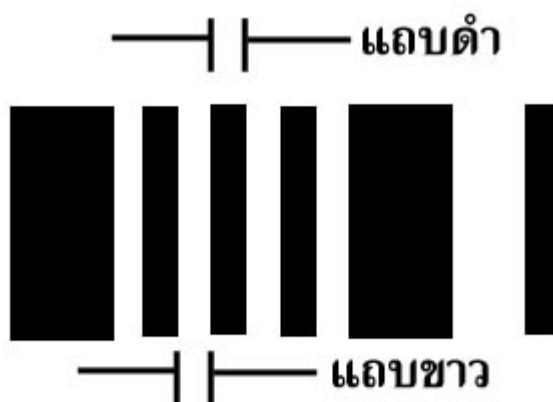


รูปที่ 2.2 เครื่องอ่านข้อมูลระบบแสงเคลื่อนที่แบบมือถือ (Hand-held moving beam)

2.2 รหัสแถบบาร์โค้ด (Barcode)

2.2.1 โครงสร้างของรหัสแถบ (The Structure of Bar Code)

รหัสแถบประกอบด้วยส่วนสำคัญสองส่วน คือ ส่วนที่เป็นแถบสีดำหรือสีเข้มโดยมีช่องว่าง (แถบสีขาว) แทรกอยู่ระหว่างกลางความกว้างของแถบสีดำและช่องว่างจะมีขนาดมาตรฐานของแต่ละชนิดของรหัสแถบ การกำหนดขนาดของแถบจะใช้แถบที่มีขนาดแคบที่สุด (X-Dimension) เป็นหลักนั่นก็คือความกว้างของแถบที่มีขนาดกว้างที่สุดนั้นจะกำหนดเป็นอัตราส่วนเทียบกับแถบที่แคบที่สุด (หนึ่งแถบแคบ) ซึ่งขนาดของแถบสีดำ (แถบที่โตที่สุด) มีขนาดเทียบกับขนาดของแถบที่แคบที่สุดเป็นอัตราส่วนเช่น 1:3 เป็นต้น ดังรูปที่ 2.3 แสดงโครงสร้างพื้นฐานของรหัสแถบ



รูปที่ 2.3 โครงสร้างพื้นฐานของรหัสแถบ

รหัสแถบบางชนิดก็มีขนาดของแถบเพียง 2 ขนาดเท่านั้น บางชนิดอาจจะมี 4 ขนาดหรือมากกว่านั้นก็ได้ แล้วแต่ชนิดของรหัสนั้นๆ การใช้รหัสแถบ ส่วนมากจะใช้แทนส่วนที่เป็นตัวเลขและส่วนที่เป็นตัวอักษรนอกจากนั้นยังมีการกำหนดตัวสัญลักษณ์พิเศษเพื่อใช้กำหนดการเริ่ม (Start) และกำหนดการสิ้นสุด (Stop) รหัสเริ่มต้นและสิ้นสุดนี้สามารถจะเปลี่ยนไปได้ตามจุดประสงค์การใช้งาน หรือเปลี่ยนตามชนิดของรหัสแถบ หรือเปลี่ยนไปตามชนิดของตัวอ่าน (Scanner) ลักษณะของรหัสแถบจะแบ่งออกเป็น 2 ลักษณะ คือแบบต่อเนื่องและแบบที่ไม่ต่อเนื่อง แบบที่ง่ายที่สุดคือแบบที่ไม่ต่อเนื่องโดยรหัสจะเริ่มต้นแถบ (Bar) และจบลงด้วยแถบโดยจะมีช่องว่าง (Gap) นี้จะเป็นช่องว่างที่ไม่มีคามหมายต่อการถอดรหัส รหัสแบบไม่ต่อเนื่อง เช่นรหัสแถบ 39 รหัสจะเริ่มต้นด้วยแถบ (Bar) แต่จะจบลงด้วยช่องว่าง (Space) หนึ่งช่อง

2.2.2 การแทนข้อมูลของรหัสแถบหลัก

รหัสแถบเป็นการแทนข้อมูลที่เป็นเลขฐานสอง (Binary Code) ในรูปของแถบขาว-ดำ ที่มีความกว้างของแถบที่ต่างกัน ในแถบดำหรือแถบขาวที่มีความกว้างจะถูกแทนค่าเป็น “1” และแถบที่มีความแคบทั้งขาวและดำจะถูกแทนค่าเป็น “0”

2.2.2.1 แถบดำที่มีความกว้างมาก เรียกว่า ไวด์บาร์ (Wide Bar)

2.2.2.2 แถบดำที่มีความกว้างน้อย เรียกว่า แนร์โรว์บาร์ (Narrow Bar)

2.2.2.3 แถบขาว (ช่องว่าง) ที่มีความกว้างมาก เรียกว่า ไวด์สเปซ (Wide Space)

2.2.2.4 แถบขาว (ช่องว่าง) ที่มีความกว้างน้อย เรียกว่า แนร์โรว์สเปซ (Narrow Space)

2.2.3 ชนิดแถบรหัส

2.2.3.1 รหัส 3 ใน 9 (3 of 9 or 39 Barcode) ในโครงการนี้ข้อมูลที่ใช้เป็นแบบตัวเลขอย่างเดียวเราจึงเลือกใช้รหัสแถบ แบบ 3 ใน 9 จึงขออธิบายเพียงชนิดเดียวเท่านั้น รหัส 3 ใน 9 อาจเรียกอีกอย่างหนึ่งว่ารหัส 39 เป็นรหัสที่ใช้แทนตัวอักษรทั้งหมด 44 ตัวเป็นอักษรภาษาอังกฤษ 26 ตัว ตัวเลข 0-9 จำนวน 10 ตัว และอักขระพิเศษอีก 8 ตัว รหัส 39 นี้ เป็นการประยุกต์ใช้รหัส 2 ใน 5 โดยนำเอาส่วนที่เป็นแถบดำ 5 แถบนั้น ประกอบด้วยแถบกว้างที่เป็นบิต “1” อยู่หนึ่งช่องและช่องว่างแคบที่เป็นบิต “0” อยู่สามช่อง ดังนั้นเมื่อรวมกันจะเป็น 9 แถบ เป็น บิต 1 อยู่สามแถบ “0” อยู่ แถบ รหัส 3 of 9 นี้มีรหัสที่เป็นรหัสเริ่มต้น (Start Code) และรหัสจบ (Stop Code) ใช้ตัวเดียวกันซึ่งมีรหัสเป็นเลขฐานสอง คือ เป็นแถบ 00110 และช่องว่าง 1000 ข้อดีข้อหนึ่งของรหัสชนิดนี้ก็คือ สามารถใช้งานได้กว้างขวางมากขึ้นเพราะสามารถใช้ตัวเลขปนกับอักษรได้และยังมีรหัสพิเศษให้เลือกใช้ได้อีก 8 ตัว คือ - . * S / + % และ ช่องว่าง

2.2.3.2 ลักษณะพิเศษของรหัส 39

ก) รหัส 39 เป็นรหัสที่ความยาวของรหัสเปลี่ยนแปลงได้ตามเครื่องอ่านข้อมูลและตามความเหมาะสมในการใช้งาน

ข) เป็นรหัสที่ตรวจสอบความผิดพลาดของข้อมูลด้วยตัวเอง (Self-Checking) ซึ่งเรียกว่า การตรวจสอบแบบพาริตี (Parity Check) เหมือนในคอมพิวเตอร์ คือมี 2 แบบ คือ แบบคี่ (Odd Parity) และแบบคู่ (Even parity) รหัส 39 นี้ใช้วิธีตรวจสอบแบบบิตคี่ โดยที่ถ้าข้อมูลมีบิต “1” เป็นคู่อยู่แล้วบิตตรวจสอบก็จะเป็นบิต “0”

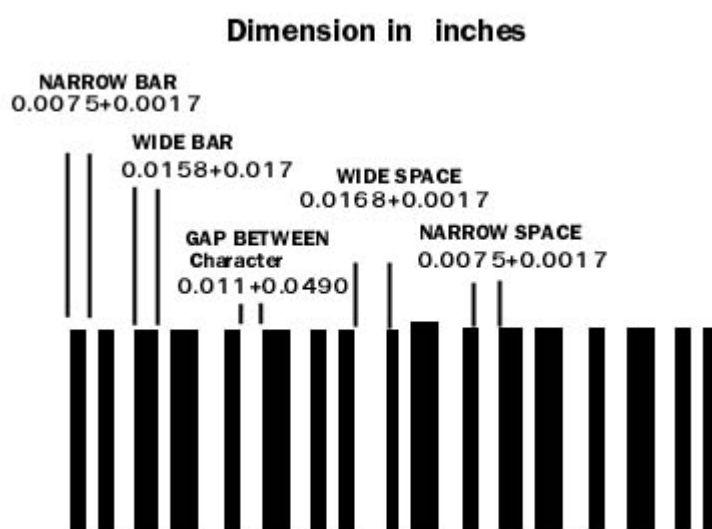
ค) รหัส 39 เป็นรหัสที่เป็นแบบไม่ต่อเนื่อง โดยจะมีช่องว่าง (Gap) อักษรที่เข้ารหัสด้วยเป็นช่องว่างที่ไม่มีความหมายต่อการถอดรหัส คือ ไม่มีข้อมูลอยู่ในช่องว่างนี้ จะต้องใส่เครื่องหมาย ! เอาไว้ด้วย ตัวอย่างเช่น “ Code!39 ” จะได้เป็น “ Code 39 ”

ง) เป็นรหัสที่มีทิศทางการอ่านได้สองทิศทาง คืออ่านจากซ้ายไปขวาหรือขวาไปซ้ายก็ได้ โดยข้อมูลจะไม่ผิดพลาด

จ) ขนาดความหนาแน่นต่อนิ้วของรหัส 39 มีหลายขนาดให้เลือกตามตาราง เพื่อใช้ให้เหมาะสมกับงานขนาดจะมีให้เลือกตั้งแต่ 9.4 ตัวอักษรต่อนิ้ว ถึง 1.7 ตัวอักษรต่อนิ้ว

ฉ) อักษรพิเศษ (รูปดอกจันที่ :*) เป็นสัญลักษณ์ของรหัสเริ่ม (Start Code) และรหัสจบ (Label) และเมื่อเครื่องอ่านถอดรหัสตัวดอกจันจะไม่ถูกถอดรหัสไปเก็บรวมไว้ในข้อมูลเช่น *code39Sample* จะได้เป็น Code39Sample

ช) ส่วนประกอบของรหัส 39 จะประกอบด้วย 5 ส่วน คือ ช่องว่างหน้ารหัส (Quiet Zones), รหัสเริ่มข้อมูล (รูปดอกจันที่), รหัสของข้อมูล, ส่วนที่รหัสสิ้นสุดข้อมูล (รูปดอกจันที่) และส่วนที่เป็นช่องว่างหลังรหัส (Quiet Zones) รูปที่ 2.4 จะแสดงโครงสร้างมาตรฐานของรหัส 39



รูปที่ 2.4 โครงสร้างมาตรฐานของรหัส 39

2.2.3.3 ขนาดมาตรฐานของรหัส 39 ขนาดความหนาแน่นของตัวอักษรต่อนิ้วนี้มีหลายขนาดให้เลือกตั้งได้ค่ามาแล้วส่วนขนาดความสูงของแถบนั้นกำหนดให้ไม่น้อยกว่า 0.24 นิ้ว หรือ 15 % ของความยาวของชุดรหัส จากนี้ยังมีขนาดที่กำหนดอีกดังนี้

ก) ขนาดของแถบที่แคบที่สุด (X- Dimension) คือ 0.0075 นิ้วหรือ 0.19 มิลลิเมตรอัตราส่วนของแถบกว้างต่อแถบแคบคือ 2 : 1 แต่อัตราส่วนจะเปลี่ยนไปเมื่อขนาดของแถบที่แคบที่สุดเปลี่ยนไป เช่น ถ้าแถบที่แคบที่สุดน้อยกว่า 0.02 นิ้ว (0.5 มิลลิเมตร) อัตราส่วนควรจะมากกว่า 2.2 :1

ข) อัตราส่วนสูงสุดควรเป็น 3:1 อัตราส่วนที่เลือกนี้จะต้องเท่ากันหมดทั้งชุดรหัส

ค) ขนาดของช่องว่างระหว่างตัวอักษร (Gap) อาจจะเป็นหนึ่งหรือ สามเท่า ของแถบที่แคบที่สุด (X-Dimension) ก็ได้หรือสูงสุดควรจะเป็น 0.06 นิ้ว (1.52 มิลลิเมตร)

ง) ช่องว่างหน้าและหลังรหัส (Leading and Trailing Quiet Zones) มีค่าต่ำสุดเท่ากับ 10 เท่าของขนาดของแถบที่แคบที่สุด (X-Dimension) หรือ 0.10 นิ้ว (2.25 มิลลิเมตร) สำหรับการกวาดอ่าน (Scan) ด้วยอาจจะได้ผลดีเมื่อมีขนาดของช่องว่างหน้าและหลังรหัส (Quiet Zone) ไม่น้อยกว่า 0.25 นิ้ว (6.4 มิลลิเมตร)

จ) ขนาดของค่าผิดพลาด (Tolerances) ในการพิมพ์แถบจะคำนวณจากสูตร

$$t = 4(N - 2)W$$

(2.1)

โดยที่ t = ขนาดของค่าผิดพลาด (Tolerance)

w = ขนาดของแถบและช่องว่างต่ำสุดเลือกจากตาราง

N = อัตราส่วนของแถบกว้างต่อแถบแคบ

2.2.3.4 ลักษณะของรหัสแถบที่ดี ควรมีคุณสมบัติดังนี้

ก) สามารถตรวจสอบความถูกต้องภายในรหัสได้

ข) ความกว้างและจำนวนของแถบรหัสจะต้องคงที่

ค) สามารถใช้แทนตัวเลขหรือตัวอักษรได้ครบ

ง) มีโครงสร้างอย่างง่าย

จ) การอ่านความเร็วที่ต่างกันควรได้ค่าที่ถูกต้องเสมอ

2.3 ประโยชน์ของรหัสแถบ

ประโยชน์รหัสแถบมีความสำคัญต่องานอุตสาหกรรมมาก และระบบอัตโนมัติที่ใช้กันมากก็คือระบบคอมพิวเตอร์ความสามารถของคอมพิวเตอร์ก็ยังมีอย่างจำกัด ไม่สามารถที่จะนำไปใช้กับทุก ๆ งานได้จากข้อจำกัดของคอมพิวเตอร์นี้จึงมีการคิดค้นที่จะนำเอาระบบรหัสแถบ (Bar Code)

มาประกอบกับเครื่องคอมพิวเตอร์เพื่อเพิ่มความสามารถที่จะทำงานได้ในขณะที่เครื่องคอมพิวเตอร์ไม่สามารถทำงานได้ แต่ระบบรหัสแถบนี้ก็ยังจะต้องมีการเชื่อมโยงกับระบบของคอมพิวเตอร์ เช่นเดียวกันไม่สามารถที่จะนำมาใช้โดดๆได้ เพราะการกำหนดสัญลักษณ์ของรหัสแถบมีการกำหนดให้มีความสัมพันธ์กับความสามารถของคอมพิวเตอร์ กล่าวคือคอมพิวเตอร์จะสามารถรับรู้ได้เพียงเลข 0 และ 1 เท่านั้น ระบบรหัสแถบจึงนำข้อจำกัดนี้มาใช้โดยใช้ รหัส 0 และ 1 ในคอมพิวเตอร์แทนด้วยแท่งขาวและดำในระบบรหัสแถบ แต่ก่อนที่แท่งขาวและดำจะเปลี่ยนไปเป็น 0 และ 1 ได้นั้นจะต้องผ่านตัวอ่านรหัสแถบที่เรียกว่าสแกนเนอร์ (Scanner) โดยต่อสแกนเนอร์เข้ากับคอมพิวเตอร์เมื่ออ่านรหัสที่เป็นแท่งขาวและดำแล้วเปลี่ยนเป็น 0 และ 1 โดยอาศัยหลักแห่งการสะท้อนแสงมากและน้อย เมื่อใช้สแกนเนอร์อ่านรหัสแถบแล้วต่อไปก็จะเป็นหน้าที่ของเครื่องคอมพิวเตอร์ที่จะประมวลผลจาก 0 และ 1 ออกใส่เป็นข้อมูลตามที่ผู้ใช้งานต้องการ เช่นวันที่ ผลิตสินค้า จำนวนที่ผลิต เป็นต้น

ความสำคัญของรหัสแถบนอกจากจะให้ความสะดวกรวดเร็วและเชื่อถือได้แล้วยังให้ความแม่นยำสูงด้วยซึ่งแทนที่จะใช้คนมาป้อนข้อมูลเพื่อบันทึกรหัสของสินค้าย่อมมีความผิดพลาดเกิดขึ้นได้และความผิดพลาดจากที่มีการสำรวจพนักงานที่มีความสามารถสูงจะความผิดพลาดประมาณ 42 % ถึง 48 % หรือ 1 ตัวอักษรต่อข้อมูลทั้ง 208-230 ตัวอักษร ดังนั้น เพื่อแก้ปัญหาของความผิดพลาดให้น้อยลง การบันทึกรหัสของสินค้าจะนำระบบรหัสแถบเข้ามาช่วย โดยเพียงแค่ใช้ตัวอ่านรหัสสแกนเนอร์ (Scanner) กวาดไปที่ตำแหน่งของรหัสแถบ การบริการจะเป็นไปอย่างรวดเร็วและความผิดพลาดที่เกิดขึ้นจากการที่นำระบบรหัสแถบมาใช้เพียง 1 ตัวอักษรต่อข้อมูลหลายล้านตัวอักษรเท่านั้น

2.4 โปรแกรม (Program) หรือซอฟต์แวร์ (Software)

การนำระบบคอมพิวเตอร์เข้ามาช่วยในการประมวลผลการทำงานของฐานข้อมูลอาจใช้โปรแกรมที่แตกต่างกันทั้งนี้ขึ้นอยู่กับระบบคอมพิวเตอร์ที่ใช้ว่าจะทำให้การใช้ฐานข้อมูลนั้นมีประโยชน์และมีประสิทธิภาพในการประมวลผลข้อมูลมากน้อยเพียงใด ไม่ว่าจะเป็นการค้นหาข้อมูลการเพิ่มข้อมูล การลบข้อมูล เป็นต้นปัญหาต่างๆที่เกิดขึ้นจากระบบการเก็บข้อมูลโดยใช้แรงงานคน ก็จะถูกแก้ไขได้ ด้วยความเร็วในการทำงานของระบบคอมพิวเตอร์ การจัดการกับข้อมูลที่มีจำนวนมากจะถูกจัดไปได้โดยใช้แผ่นดิสก์เพียงไม่กี่แผ่นซึ่งการประมวลผลฐานข้อมูลสามารถทำได้ทั้งหมด โดยใช้วิธีการที่เรียกว่า "การจัดการฐานข้อมูล" (Database Management System :

DBMS) ซึ่งเป็นโปรแกรมหรือซอฟต์แวร์ที่ทำหน้าที่ในการจัดการฐานข้อมูล โดยเป็นสื่อกลางการติดต่อระหว่างผู้ใช้ (User) กับคอมพิวเตอร์

2.5 ฐานข้อมูล

ฐานข้อมูลประกอบด้วยรายละเอียดของข้อมูลที่เกี่ยวข้องกัน ซึ่งถูกนำมาใช้ใน งานด้านต่างๆ เช่น ข้อมูลการให้สินเชื่อหรือด้านการศึกษา จะมีฐานข้อมูลที่ประกอบไปด้วย รหัสนักศึกษา ชื่อ นามสกุล ที่อยู่ เบอร์โทรศัพท์ เป็นต้น ข้อมูลเหล่านี้จะถูกเก็บไว้อย่างมีระบบ เพื่อประโยชน์ในการจัดการและการเรียกใช้ข้อมูลอย่างมีประสิทธิภาพ

2.6 โครงสร้างข้อมูล

โครงสร้างของข้อมูลสามารถเขียนให้อยู่ในรูปของตารางซึ่งจะประกอบไปด้วยแถวตามแนวตั้งและแถวตามแนวนอนโดยที่ข้อมูลนั้นมีความสัมพันธ์กันดังรูปที่ 2.5

ชื่อ	อายุ	เพศ	จังหวัด
พิเศษ งามรุ่งโรจน์	19	ชาย	อ่างทอง
เกษม สุธรรมมา	20	ชาย	เชียงใหม่

รูปที่ 2.5 ฐานข้อมูลเบื้องต้น

2.6.1 ฟیلด์ (Field) คือแถวตามแนวตั้ง โดยในแต่ละฟیلด์ จะมีข้อมูลอยู่ประเภทเดียว เช่น ฟیلด์ของอายุก็จะมีแต่ข้อมูลของอายุเท่านั้น ดังในตัวอย่างนี้จะมีทั้งหมด 4 ฟیلด์ และในแต่ละฟیلด์ จะมีส่วนขยายว่าฟیلด์นั้นเป็นฟیلด์ของอะไร เช่น ฟิลด์ที่ใช้เก็บชื่อเราจะตั้งให้ Field Name มีชื่อว่า Name

2.6.2 เรคอร์ด (Record) คือแถวตามแนวนอน เป็นการนำฟิลด์หลายฟิลด์มารวมกันเพื่อให้ข้อมูลได้สื่อความหมายตามต้องการ หลักการในการสังเกตเรคอร์ดง่ายๆ โดยดูจากข้อมูลของคน 1 คน ก็คือจะมี 1 เรคอร์ด

2.6.3 ไฟล์ (File) ไฟล์หรือแฟ้มข้อมูลจะเป็นการนำเอาเรคอร์ดหลายๆเรคอร์ดที่มีความสัมพันธ์กัน มารวมกัน ในกรณีที่น่าข้อมูลเหล่านี้มาเขียนในบัตร

2.7 ระบบฐานข้อมูล

ระบบฐานข้อมูล คือ การนำข้อมูลในองค์การที่มีความเกี่ยวข้องกันจัดเก็บไว้อย่างเป็นระบบในที่เดียวกันโดยที่ผู้ใช้ฐานข้อมูลแต่ละคนจะมองข้อมูลนี้ในแง่มุมที่แตกต่างกันไปตามจุดประสงค์ของการประยุกต์ใช้งาน นอกจากนี้การใช้ระบบฐานข้อมูลยังได้อำนวยความสะดวกให้แก่ผู้ใช้ เพราะผู้ใช้ไม่จำเป็นต้องสนใจว่าลักษณะการจัดเก็บข้อมูลโดยแท้จริงแล้วเป็นเช่นไร นั่นก็คือในระดับการมองของผู้ใช้ ก็เพียงมุ่งหวังให้เรียกใช้ข้อมูลในงานของเขามีประสิทธิภาพมากที่สุดเท่านั้น ซึ่งแน่นอนว่าการที่จะให้การใช้ข้อมูลมีประสิทธิภาพมากที่สุดเท่านั้น ย่อมจะเป็นสืบเนื่องมาจากการออกแบบ และเลือกเฟ้นวิธีการจัดเก็บข้อมูลที่เหมาะสมซึ่งก็เป็นเรื่องที่ยุ่งยาก แต่ระบบก็ได้ซ่อนรายละเอียดเหล่านี้จากผู้ใช้ การซ่อนรายละเอียดเช่นนี้เป็นไปได้ ก็เพราะระบบฐานข้อมูลได้จัดแบ่งระดับข้อมูลออกเป็น 3 ระดับ คือ

2.7.1 ระดับภายใน (Internal หรือ Physical Level) เป็นระดับที่ต่ำที่สุด อันได้แก่ระดับของการจัดเก็บข้อมูลจริงๆ ซึ่งเนื้อหาในส่วนนี้เกี่ยวข้องกับ โครงสร้างของข้อมูล

2.7.2 ระดับหลักการ (Conceptual Level) เป็นระดับที่อยู่ถัดขึ้นมา อันได้แก่ระดับของการมอง และความสัมพันธ์ระหว่างข้อมูลทั้งหมดรวมทั้งกฎเกณฑ์ต่างๆ เกี่ยวกับข้อมูลและผู้มีสิทธิ์ที่จะใช้

2.7.3 ระดับภายนอก (External หรือ View Level) เป็นระดับที่อยู่สูงสุดอันเป็นระดับข้อมูลที่จะมองเห็นจากการใช้งานของผู้ใช้แต่ละคน

2.8 ระบบจัดการฐานข้อมูล (Database Management System : DBMS)

ระบบรายงานคอมพิวเตอร์ทุกระบบในปัจจุบันต้องการขบวนการที่จะเข้าถึง (Access) ข้อมูลที่ต้องการได้อย่างสะดวกรวดเร็ว ดังนั้นจึงต้องพยายามออกแบบฐานข้อมูล (Database) ให้เกิดความสะดวกและความซับซ้อนกันของข้อมูลให้ได้มากที่สุด เพื่อลดความยุ่งยากในการบำรุงรักษาฐานข้อมูล การใช้ฐานข้อมูลจึงเริ่มมีบทบาทมากและค่อยๆมาแทนที่เพิ่มข้อมูลแบบมาตรฐาน

ระบบจัดการฐานข้อมูล คือระบบโปรแกรมที่มีความสามารถในการจัดการข้อมูลในด้านต่างๆ ได้แก่ การให้คำจำกัดความของข้อมูลและเรคอร์ด การกำหนดความสัมพันธ์ระหว่างฟิลด์ต่าง ๆ ในเรคอร์ด การจัดการประมวลผล ปรับเปลี่ยนแก้ไขข้อมูล และจัดการกำหนดควบคุมการใช้ข้อมูลที่มีอยู่ได้อย่างเป็นระบบ จุดมุ่งหมายสำคัญของระบบจัดการฐานข้อมูล (DBMS) สามารถจำแนกออกได้เป็น 2 ด้าน คือเพื่อการควบคุมและเพื่อสนับสนุนการใช้ข้อมูลภายในองค์การอย่างเป็นระบบ

2.9 ประโยชน์จากการประมวลผล

2.9.1 ลดความซ้ำซ้อนของข้อมูล การประมวลผลโดยใช้ไฟล์ธรรมดานั้นจำเป็นที่ผู้ใช้ แต่ละกลุ่มจะต้องมีไฟล์ส่วนตัวเอาไว้ ดังนั้นจึงเกิดเหตุการณ์ที่ข้อมูลชนิดเดียวกันถูกเก็บไว้หลายๆแห่ง หรือที่เรียกกันว่าความซ้ำซ้อน การนำข้อมูลทั้งหมดมาเก็บไว้ที่เดียวกันในฐานข้อมูลนี้เป็นการลดความซ้ำซ้อนลงไปได้

2.9.2 สามารถหลีกเลี่ยงความขัดแย้ง ของข้อมูลได้ในระดับหนึ่ง การเก็บข้อมูลไว้หลายๆ แห่ง อาจจะก่อให้เกิดปัญหาว่า การแก้ไขข้อมูลเดียวกันนี้ทำไม่เหมือนกันในทุกๆแห่ง ทำให้ข้อมูลชุดเดียวกันนี้อาจมีค่าในแต่ละแห่งไม่ตรงกัน

2.9.3 สามารถใช้ข้อมูลร่วมกันได้ ไม่จำกัดอยู่เฉพาะ โปรแกรม ที่ใช้ข้อมูลอยู่ในปัจจุบันเท่านั้น แต่ยังสามารถไปถึง โปรแกรมประยุกต์ที่จะพัฒนาขึ้นมาใหม่ด้วยที่จะสามารถใช้ข้อมูลที่มีอยู่ได้เลย

2.9.4 สามารถควบคุม เป็นมาตรฐานได้ การที่เรานำข้อมูลมาเก็บรวมกันไว้ในฐานข้อมูล ทำให้ผู้ที่หน้าที่ควบคุมดูแลการใช้ระบบฐานข้อมูลสามารถกำหนดมาตรฐานของข้อมูลขึ้นมาได้

2.9.5 สามารถจัดหาระบบความปลอดภัยที่รัดกุม คือสามารถป้องกันไม่ให้ผู้ใช้ที่ไม่มีสิทธิ์ มาใช้ข้อมูลในระบบได้ และยังสามารถกำหนดสิทธิ์การใช้ให้แก่ผู้ใช้คนใดๆ ก็ได้ตามความเหมาะสม และผู้ใช้แต่ละคนก็อาจจะใช้ข้อมูลได้ในระดับที่ต่างกัน

2.9.6 สามารถควบคุมความคงสภาพของข้อมูลได้ คือผู้ออกแบบสามารถใช้กฎเกณฑ์เพื่อควบคุมความคงสภาพไว้ได้ เพื่อให้ข้อมูลถูกต้องตามกฎเกณฑ์ที่กำหนด

2.9.7 สามารถสร้างสมดุลในความขัดแย้งของความต้องการได้ เช่น เลือกเก็บข้อมูลที่จะต้องใช้บ่อยๆไว้ในสื่อข้อมูลที่มีความเร็วเป็นพิเศษ

2.9.8 เกิดความเป็นอิสระของข้อมูล

2.10 ขั้นตอนการออกแบบระบบฐานข้อมูล

2.10.1 รวบรวมข้อมูลในการออกแบบระบบฐานข้อมูลผู้ออกแบบจะต้องเข้าใจระบบที่ต้องการสร้างทั้งระบบก่อนต้องรู้ว่าข้อมูลอะไรบ้าง อยู่ในระบบและต้องการผลลัพธ์ใดบ้าง ดังนั้นผู้ออกแบบจะต้องรวบรวมข้อมูลต่างๆทั้งหมดที่มีอยู่ในระบบจากผู้ที่เกี่ยวข้องแต่ละฝ่ายเข้ามาเป็นผู้ให้ข้อมูลดังกล่าว วิธีการที่ใช้รวบรวมข้อมูลมีหลายวิธี เช่น การใช้แบบสอบถาม , สัมภาษณ์และสังเกต การปฏิบัติงานของผู้ใช้หรือ วิธีการประชุมสัมมนาาร่วมกัน เป็นต้น สิ่งที่ผู้ออกแบบจะต้องศึกษารวบรวมข้อเท็จจริงของระบบ ประกอบด้วยกฎเกณฑ์การสร้างข้อมูล กฎเกณฑ์การเรียกใช้ข้อมูล วิธีการ

จัดการข้อมูล กฎเกณฑ์การตรวจสอบความถูกต้องของข้อมูล และกฎเกณฑ์การยกเลิกการใช้ข้อมูล ชุดนั้นๆ

2.10.2 ทำความเข้าใจกับข้อมูล วิธีการที่นิยมใช้กันมากในการศึกษาเพื่อวางรูปแบบระบบฐานข้อมูล คือวิธีการแยกแยะความสัมพันธ์ (Factorial) ซึ่งแบ่งได้เป็น 4 ลักษณะ

2.10.2.1 จุดมุ่งหมายของฐานข้อมูล (Task-Data Relationship) จุดมุ่งหมายของฐานข้อมูลคืออะไร ชิ้นงานผลลัพธ์ที่ต้องการคืออะไร จะต้องใช้ข้อมูลใดบ้างในการที่จะได้ผลลัพธ์ดังกล่าว

2.10.2.2 การจัดข้อมูลในฟิลด์ต่างๆ (Field-Field / Key Field / Key – Key Relationship) การจัดข้อมูลเป็นฟิลด์ต่างๆนั้นมีฟิลด์ มีฟิลด์ใดเป็นฟิลด์หลัก(Key) ในการกำหนดรูปแบบหรือลักษณะของเรคอร์ด และถ้ามีมากกว่า 1 ฟิลด์หลัก แต่ละฟิลด์จะมีความสัมพันธ์กันอย่างไร

2.10.2.3 การจัดกลุ่มฟิลด์ที่สัมพันธ์กันเป็นเรคอร์ด (Field-Record Relationship) ลักษณะการจัดกลุ่มฟิลด์ที่สัมพันธ์กันเป็นเรคอร์ดเป็นไปในลักษณะใด ทำอย่างไรจึงไม่ซ้ำซ้อน

2.10.2.4 การเชื่อมโยงความสัมพันธ์ระหว่างไฟล์ (File-File Relationship) ถ้าแยกข้อมูลเก็บไว้หลายๆไฟล์ จะเชื่อมโยงความสัมพันธ์ระหว่างไฟล์ได้อย่างไร การทำความเข้าใจกับระบบมีประเด็นต่างๆต่อไปนี้ คือ

- ก) แหล่งข้อมูล (จะรวบรวมข้อมูลเพื่อจัดเก็บและใช้งานในฐานข้อมูลได้จากที่ใด)
- ข) ค่าของข้อมูลเหล่านี้ในฐานข้อมูลนานเท่าใด เมื่อใดจะลบทิ้งได้
- ค) ข้อมูลแต่ละตัวแต่ละชุด มีความสัมพันธ์กันกับข้อมูลตัวอื่นๆ ได้อย่างไร โดยใช้วิธีการแยกแยะความสัมพันธ์
- ง) ข้อมูลเหล่านี้ทำให้เกิดการสารสนเทศอะไรได้บ้าง และอย่างไร
- จ) กระบวนการแปรผลของข้อมูลเป็นอย่างไร ค่าจำกัดความของข้อมูลแต่ละตัวคืออะไร
- ฉ) ข้อมูลเหล่านี้มีโอกาสเปลี่ยนแปลงบ่อยแค่ไหนอย่างไร
- ช) ข้อมูลเหล่านี้จะถูกจัดเก็บอย่างปลอดภัย ไม่ทำลายโดยอุบัติเหตุหรือสาเหตุอื่นๆได้อย่างไร
- ซ) มีกฎเกณฑ์ในการตรวจสอบความถูกต้องของข้อมูลได้อย่างไรบ้าง

2.10.3 การกำหนดรูปแบบระบบขั้นต้น (Initial Model)

ขั้นตอนถัดมาหลังจากที่ทีมงานตกลงกันเรียบร้อยแล้วในเรื่องฟิลด์ และตาราง (Table) ที่จะใช้ภายในระบบ ทีมงานจะต้องกำหนดร่างของระบบคร่าวๆว่า ภายในระบบจะมีการส่งผ่านข้อมูลและแสดงผลลัพธ์ (Input –Output) ในลักษณะใด

2.10.4 วิเคราะห์รายละเอียดและลงมือสร้างระบบ

หลังจากที่ได้รูปแบบขั้นต้นแล้ว จะต้องศึกษาอย่างละเอียดว่าจะต้องมีการจัดการอย่างตาม ลำดับขั้นตอน แล้วจึงวาดรูปแบบของระบบในรายละเอียดเป็นขั้นตอนสุดท้ายขั้นตอนการวิเคราะห์รายละเอียดของระบบที่สร้างนี้เป็นขั้นตอนที่สำคัญ เป็นการกำหนดว่ามีการใช้ตาราง (Table) อย่างไร และมีขั้นตอนในการดำเนินงานของระบบในภาพรวมอย่างไรทั้งนี้เพื่อที่จะเป็นแนวทางสำหรับผู้ที่จะลงมือสร้างระบบต่อไป

2.11 ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

ฐานข้อมูลเชิงสัมพันธ์เป็นระบบจัดการฐานข้อมูล (Database Management System : DBMS) รูปแบบหนึ่งที่นิยมใช้ปัจจุบัน เป็นระบบฐานข้อมูลที่ประกอบไปด้วยตาราง (Table) หลายๆ ตาราง ที่มีความสัมพันธ์ ระหว่างกัน

ในตารางแต่ละตารางจะต้องมีคีย์ (Key) ที่ชี้ไปยังข้อมูลแต่ละเรคอร์ดสามารถแยกออกได้เป็น 2 ชนิด ได้แก่

ก) ไพรมารีคีย์ (Primary Key) เป็นฟิลด์ (Field) หรือกลุ่มของฟิลด์ ที่มีไว้สำหรับอ้างอิงถึงข้อมูลแต่ละแถวคุณสมบัติสำคัญของไพรมารี คีย์ คือในแถวแต่ละแถวของตาราง (Table) จะไม่มีฟิลด์ที่ซ้ำกัน

ข) ฟอเรนจ์คีย์ (Foreign Key) เป็นฟิลด์ ที่ไม่ได้เป็นไพรมารีคีย์ ของตาราง (Table) ปัจจุบัน แต่เป็นไพรมารีคีย์ ของตาราง (Table) อื่น

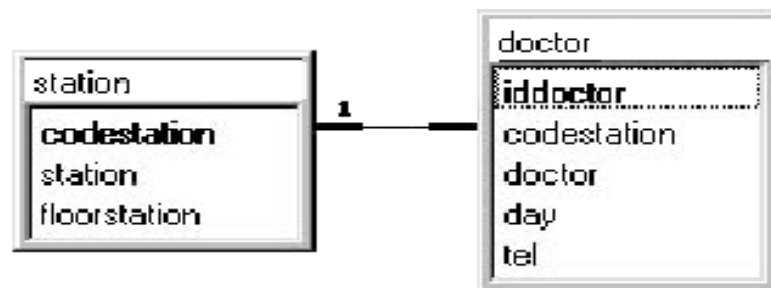
2.11.1 หลักการกำหนดความสัมพันธ์แต่ละตาราง (Table)

จุดมุ่งหมายของการกำหนดความสัมพันธ์ คือ การกำหนดรูปแบบและวิธีการเข้าถึงและการเรียกใช้ (Access) ข้อมูลในเรคอร์ดที่อยู่ต่างไฟล์กันในขั้นแรกผู้ออกแบบจะต้องมองดูความสัมพันธ์ด้วยสายตาของผู้ใช้ระบบ คือ พิจารณาว่าผู้ใช้ต้องการข้อมูลหรือผลลัพธ์อะไรบ้าง หลังจากนั้น จึงพิจารณาด้วยสายตาของผู้เขียนโปรแกรมว่าข้อมูลที่ต้องการนั้นจัดในเรคอร์ดใดไฟล์ใดและจะเอาข้อมูลที่จัดเก็บในที่ต่าง ๆ นั้นมาประสานกันออกเป็นผลลัพธ์ที่ต้องการได้อย่างไร

ในการกำหนดความสัมพันธ์ระหว่างตาราง (Table) ก่อนอื่น จะต้องมีการกำหนด ฟิลด์ข้อมูลร่วม (Common Field) ซึ่งเป็นฟิลด์ที่มีอยู่ในทั้งสองตาราง

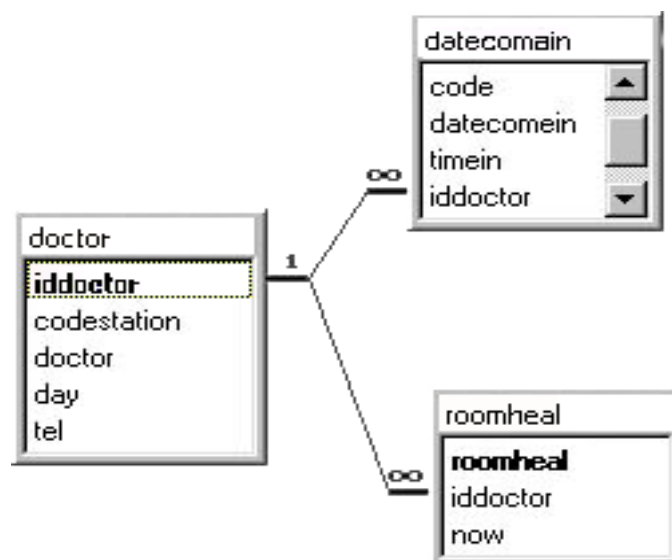
การกำหนดความสัมพันธ์ระหว่างไฟล์หรือตาราง (Table) สามารถทำได้ใน 3 ลักษณะดังนี้

2.11.1.1 ความสัมพันธ์แบบหนึ่งฟิลด์ต่อหนึ่งฟิลด์ (One - to - One) จากรูปที่ 2.6 เป็นตัวอย่างตารางที่ความสัมพันธ์แบบ หนึ่งต่อหนึ่ง ในกรณีนี้ใช้ Codestation เป็นฟิลด์ข้อมูลร่วม (Common Field) เพื่อเชื่อมโยงข้อมูลในตาราง



รูปที่ 2.6 ความสัมพันธ์แบบหนึ่งฟิลด์ต่อหนึ่งฟิลด์ (One to One)

2.11.1.2 ความสัมพันธ์แบบหนึ่งฟิลด์ต่อหลายฟิลด์ (One - to - many) จากตัวอย่างในรูปที่ 2.7 เป็นความสัมพันธ์แบบหนึ่งฟิลด์ต่อหลายฟิลด์ โดยมีฟิลด์ iddoctor เป็นฟิลด์ข้อมูลร่วม (Common Field) จากตัวอย่างนี้ รหัสประจำตัวของแพทย์หนึ่งคนมีความสัมพันธ์กับทั้งตาราง datecomein และตาราง roomheal



รูปที่ 2.7 ความสัมพันธ์แบบหนึ่งฟิลด์ต่อหลายฟิลด์ (One-to-many)

2.11.1.3 ความสัมพันธ์แบบใช้เชื่อมข้อมูลหลายเรคอร์ดในตารางหนึ่งโดยที่เชื่อมโยงไป อีกหลายเรคอร์ด (Many - to - Many)

2.11.2 กระบวนการจัดการออกแบบโครงสร้างข้อมูล (Normalization)

กระบวนการจัดการออกแบบโครงสร้างข้อมูลที่ละขั้นตอน ทั้งนี้เพื่อกำจัดปัญหาที่อาจจะเกิดขึ้นในขณะที่ผู้ใช้ส่งแก้ไข (Update) หรือเรียกใช้งาน (Access) ข้อมูลที่อยู่ในเรคอร์ดต่างๆในไฟล์ ในฐานะของผู้ออกแบบระบบที่ต้องการกำหนดโครงสร้างของข้อมูล (กำหนดฟิลด์ต่างๆในเรคอร์ด) ผู้ออกแบบต้องกำหนดวิธีการที่จะใช้วิธีการที่จะดำเนินการเพื่อ หลีกเลี่ยงการกำเนิดปัญหาในการใช้และจัดการกับข้อมูลเมื่อนำระบบมาติดตั้งใช้งานจริง

รูปแบบการจัดข้อมูลที่ไม่ทำให้การมีอยู่ของหน่วยใดหน่วยหนึ่ง มีอิทธิพลต่อการมีอยู่ของอีกหน่วยหนึ่งเสมอไป (Insertion Anomaly) ตัวอย่างเช่น ผู้ออกแบบระบบเห็นว่าไม่จำเป็นต้องมีตาราง (Table) เกี่ยวกับผลิตภัณฑ์สินค้าของบริษัท (Product Table) และ ราคา (Price) ไปเก็บรวบรวมไว้ในไฟล์เกี่ยวกับรายการสั่งซื้อสินค้าของลูกค้า (Line item Table) ดังนั้นเมื่อบริษัทผลิตสินค้าตัวใหม่ และต้องการเก็บข้อมูลเกี่ยวกับสินค้าใหม่ ไว้ในฐานข้อมูลก็จะต้องไปเพิ่ม เรคอร์ดในไฟล์เกี่ยวกับรายการสั่งซื้อสินค้าของลูกค้า (Line item Table) แต่การเพิ่มเรคอร์ดในไฟล์เกี่ยวกับรายการสั่งซื้อสินค้าของลูกค้า (Line item Table) จะกระทำไม่ได้ก็ต่อเมื่อมีลูกค้าสั่งซื้อสินค้าเท่านั้น

รูปแบบการตั้งลบเรคอร์ดหนึ่งแล้วอาจมีผลกระทบไปอีกเรคอร์ดหนึ่งที่ไม่ได้ตั้งใจจะลบไปด้วย (Deletion Anomaly) เป็นรูปแบบการจัดที่ตรงข้ามกับแบบแรก ในตัวอย่างเดิมถ้าผู้ใช้ตั้งลบเรคอร์ดของลูกค้าที่สั่งซื้อสินค้าตัวใดตัวหนึ่งเข้ามา ก็อาจจะเป็นการลบข้อมูลเกี่ยวกับสินค้าตัวนั้นออกไปจากฐานข้อมูล ทั้งนี้เพราะรายละเอียดเกี่ยวกับสินค้าตัวนั้นถูกจัดเก็บรวบรวมอยู่กับ ฟิลด์ต่างๆของการสั่งซื้อสินค้าในรายการสั่งซื้อสินค้าของลูกค้า (Line item Table)

การเปลี่ยนแปลงค่าฟิลด์ใดๆในฐานข้อมูลจะมีการเปลี่ยนแปลงเฉพาะในฟิลด์เดียวในระบบฐานข้อมูลทั้งหมด ถ้าค่าเดียวจะต้องถูกเปลี่ยนในหลายๆที่รูปแบบ (Update Anomaly) ตัวอย่างเช่นในรายการสั่งซื้อสินค้าของลูกค้า (Line item Table) เดิมถ้ามีรายการ 25 รายการ ที่สั่งซื้อสินค้าตัวหนึ่งเข้ามา ดังนั้นก็จะมีทั้งหมด 25 เรคอร์ด ใน Table เกี่ยวกับสินค้าตัวนั้น แต่ต่อมาผู้ใช้ระบบพบว่ารายการสินค้าตัวนั้นมีชื่อระบุผิดมาตลอด ดังนั้นเมื่อจะแก้ไขชื่อสินค้าก็จะต้องส่งแก้ทั้งหมดใน 25 เรคอร์ด แทนที่จะส่งแก้ที่เดียว

ลักษณะการซ้ำซ้อนของข้อมูลนั้น ไม่ควรจะจัดให้มีอยู่ในรูปแบบโครงสร้างใดๆตั้งแต่ช่วงแรกของการออกแบบ แต่อย่างไรก็ตาม จุดที่ยอมรับให้มีการซ้ำซ้อนของฟิลด์ได้ก็คือฟิลด์ที่ทำหน้าที่เป็นฟอเรนจี้คีย์ (Foreign Key) ที่เชื่อมโยงความสัมพันธ์ระหว่างข้อมูลระหว่างตาราง (Table)

ต่างๆกันได้ การป้องกันและการควบคุมความซ้ำซ้อน เป็นปัจจัยสำคัญในการออกแบบระบบฐานข้อมูล และแก้ไขปัญหานี้สามารถทำได้โดยผ่านกระบวนการจัดการออกแบบโครงสร้างข้อมูล

กระบวนการจัดการออกแบบโครงสร้างข้อมูล (Normalization) เริ่มต้นด้วยการพิจารณาจากมุมมองของผู้ใช้ระบบได้แก่

2.11.2.1 รูปแบบโครงสร้างข้อมูลลำดับที่ 1 (First Normal Form : 1NF) เป็นรูปแบบของโครงสร้างข้อมูลในแต่ละตารางไม่มีฟิลด์ที่เป็น รีพีตติ้งกรุป (Repeating Group) ถ้าหากว่าในตารางมีรีพีตติ้งกรุป ถ้าหากว่าในตารางมีรีพีตติ้งกรุป จะต้องแยกรีพีตติ้งกรุป นั้นออกมาสร้างเป็นตารางใหม่ โดยเพิ่มไพรมารีคีย์ (Primary Key) ของตารางเดิมมาเป็นไพรมารีคีย์ ของตารางเดิมมาเป็นไพรมารีคีย์ ใหม่ร่วมกับไพรมารีคีย์ ของตารางใหม่ที่สร้างขึ้น

2.11.2.2 รูปแบบโครงสร้างข้อมูลลำดับที่ 2 (Second Normal Form : 2NF) เป็นโครงสร้างข้อมูลที่มีข้อมูลแต่ละตัวสามารถอ้างอิงโดยใช้ ไพรมารีคีย์ (Primary Key) ทั้งหมดของตารางนั้น (ในกรณีที่มีไพรมารีคีย์มากกว่า 1 ตัว) ถ้าหากว่ามีฟิลด์ใดฟิลด์หนึ่งที่สามารถอ้างอิงได้โดยการใช้เพียงส่วนหนึ่งส่วนใดของ ไพรมารีคีย์ ทั้งหมด

2.11.2.3 รูปแบบโครงสร้างข้อมูลลำดับที่ 3 (Third Normal Form : 3NF) เป็นโครงสร้างข้อมูลในตารางที่จะมีฟิลด์ใดฟิลด์หนึ่งอ้างอิงได้โดยขึ้นอยู่กับฟิลด์ที่ไม่เป็นไพรมารีคีย์ (Primary Key) ของตารางไม่ได้ ถ้าหากว่ามีเหตุการณ์ดังกล่าวเกิดขึ้น ให้แยกข้อมูลดังกล่าวมาสร้างเป็นตารางใหม่ และกำหนดให้ ตัวที่เป็นไพรมารีคีย์ ที่ตารางใหม่ เป็นฟอเรนคีย์ (Foreign Key) ในตารางเก่า

2.12 ไมโครซอฟท์แอคเซส (Microsoft Access)

ไมโครซอฟท์แอคเซส (Microsoft Access) เป็นโปรแกรมฐานข้อมูลของผู้พัฒนาซึ่งง่ายพอที่ ผู้ใช้ จะเรียนรู้จัดการเองได้ ปัจจุบันมีการปรับปรุงที่สำคัญจากเวอร์ชันก่อนและเพิ่มความสามารถในอินเทอร์เน็ตสำหรับผู้ใช้งาน และเรื่องของความสัมพันธ์

โปรแกรมไมโครซอฟท์แอคเซส เป็นโปรแกรมจัดการฐานข้อมูล(DBMS) แบบฐานข้อมูลสัมพันธ์ (Relational Database) ซึ่งมีส่วนประกอบในการสร้างแอปพลิเคชัน (Application) ดังต่อไปนี้

2.12.1 ตาราง (Table)

ตาราง คือ ไฟล์ข้อมูลต่างๆซึ่งประกอบด้วยเรคอร์ดต่างๆ การสร้างตารางในขั้นต้น เริ่มจากการสร้างโครงสร้างของตารางขึ้นมาก่อน นั่นคือการตั้งชื่อฟิลด์ ชนิดของข้อมูล (Data Type) และขนาดความยาวของฟิลด์ ภายในเรคอร์ด ต่อจากนั้นจึงทำการกรอกข้อมูลต่างๆลงในฟิลด์

2.12.2 ฟอรั่ม (Form)

ฟอรั่มในระบบฐานข้อมูล หมายถึง รูปแบบในการกรอกข้อมูลต่างๆลงในฟิลด์โดยผ่านทางจอ ซึ่งให้อำนวยความสะดวกในการติดต่อกับระบบแก่ผู้ใช้โดยการปรับปรุงการกรอกข้อมูลให้ง่ายขึ้น และสะดวกต่อการนำมาใช้งานมากกว่าการป้อนข้อมูลในตารางโดยตรง โดยแสดงข้อมูลเพียงบางส่วนที่ผู้ใช้จำเป็นต้องใช้เท่านั้น โดยแหล่งข้อมูลที่สัมพันธ์กับฟอรั่มจะได้มาจากตาราง หรือ คิวรี (Query) ก็ได้

2.12.3 รายงาน (Report)

รายงาน คือ ข้อมูลต่างๆที่ถูกพิมพ์ออกมาหรือถูกแสดงบนจอภาพ เพื่อแสดงข้อมูลส่วนที่ต้องการทราบจากฐานข้อมูล ซึ่งข้อมูลที่ต้องการนั้นอาจผ่านการคำนวณมาก่อนแล้วก็ได้มาจากตาราง หรือคิวรี เช่นเดียวกับ ฟอรั่ม

2.12.4 คิวรี (Query)

คิวรีเป็นวิธีการในการเลือกสรรข้อมูลชนิดต่างๆตามชนิดของเรคอร์ดในตารางตาม que ที่ผู้ใช้ต้องการ สามารถสร้างคิวรี โดยติดต่อกับหน้าจอ (Graphics User Interface) หรือผ่านการใช้ภาษาเอสคิวแอล (Structure Query Language : SQL) ส่วนการสร้างแอปพลิเคชัน (Application) ในไมโครซอฟท์แอกเซสสามารถทำโดยผ่านวิธีการ 2 วิธีได้แก่

2.12.4.1 การใช้มาโคร (Macro) เป็นวิธีการทำงานโดยอัตโนมัติ ที่รวมเอาการทำงานหลายๆอย่างเอาไว้ ขั้นตอนเพื่อให้เกิดผลลัพธ์ที่ต้องการมารวมกันไว้ในทีเดียว เพื่อให้สะดวกต่อการใช้งาน การสร้างมาโคร (Macro) สามารถทำได้โดยการผ่านมาโครเอดิเตอร์ (Macro Editor) ของไมโครซอฟท์แอกเซสซึ่งการสร้างมาโครนี้ทำได้ง่าย

2.12.4.2 การใช้แอกเซสเบสิก (Access Basic) เป็นวิธีการเขียนโปรแกรมเพื่อสร้างลำดับการทำงานของฟอรั่ม หรือเพื่อดึงเอาข้อมูลจากฐานข้อมูลมาใช้งาน และสร้างเอาท์พุท

2.12.5 นิพจน์

นิพจน์หรือการอ้างถึงการกำหนดค่าในไมโครซอฟท์แอกเซส ใช้กับการสร้างงานทุกชนิดในระบบฐานข้อมูล นิพจน์ประกอบด้วยส่วนประกอบ 5 แบบด้วยกันคือ

2.12.5.1 ตัวอักษร เราสามารถใช้ตัวอักษรพิเศษที่เรียกว่าเดลิมิเตอร์ (Delimiter) ก่อนและหลังจากตัวอักษรเพื่อบอกถึงประเภทของข้อมูล ในตาราง แสดงถึงประเภทของตัวอักษรที่สามารถถูกใช้ในนิพจน์

ตารางที่ 2.1 การป้อนตัวอักษรลงในนิพจน์

ประเภท	เดลิมิเตอร์	ตัวอย่าง
ข้อความ	"	"ขอบคุณ!"
วันที่	#	# 9/1/94 #
ฟิลด์หรือคอนโทรล	[]	[Date of Birth]

2.12.5.2 ตัวปฏิบัติการ ในการกำหนดนั้นตัวปฏิบัติการจะบอกชนิดของการกระทำที่วิธีกำหนดค่า นั้นต้องทำ หรือการกำหนดค่าเปรียบเทียบค่า 2 ค่า

ก) ตัวปฏิบัติการแบบพีชคณิตและข้อความ ตัวปฏิบัติการแบบพีชคณิตจะคำนวณในเชิงตัวเลขหรือทางคณิตศาสตร์ ยกตัวอย่างในการกำหนดค่าผลคูณของค่าในฟิลด์ที่มีชื่อว่า Sales Tax Rate กับฟิลด์หรือส่วนควบคุมที่ชื่อว่า Order Subtotal

[Sales Tax Rate] [Order Subtotal]

ได้มีตัวปฏิบัติการทางพีชคณิตสำหรับการบวก ซึ่งใช้สำหรับการเชื่อมข้อความเข้าด้วยกัน นั่นคือตัวปฏิบัติการ & ยกตัวอย่างเช่น ใช้การกำหนดค่าเพื่อจำหน่ายของจดหมาย

" Dear " & [First Name] & " ; "

โดยที่ [First name] เป็นชื่อฟิลด์ในตารางคิวรี

ตารางที่ 2.2 ตัวปฏิบัติการทางพีชคณิต

ตัวปฏิบัติการ	หน้าที่
^	ยกกำลังตัวเลข
*	การคูณตัวเลข
/	การหารตัวเลข
\	การหารตัวเลข โดยให้ผลลัพธ์ในส่วนเฉพาะจำนวน
+	การบวกตัวเลข
-	การลบตัวเลข 2 จำนวนหรือตัวเลขที่มีค่าติดลบ
Mod	การหารตัวเลขโดยให้ผลลัพธ์ในรูปเศษ

&	การเชื่อมข้อความ 2 ข้อความ
---	----------------------------

ข) ตัวปฏิบัติการแบบเปรียบเทียบ ตัวปฏิบัติการแบบเปรียบเทียบค่านั้นยกตัวอย่าง เช่น $> = \# / 1 / 1 / 93 \#$ ใช้ในช่อง Criteria ของ QBE ในคิวรี เพื่อแยกเรคอร์ดสำหรับวันที่ที่มากกว่า (หลังจาก) หรือเท่ากับวันที่ 1 มกราคม 1993 สรุปตัวปฏิบัติการแบบเปรียบเทียบ

ตารางที่ 2.3 ตัวปฏิบัติการแบบเปรียบเทียบ

ตัวปฏิบัติการ	ความหมาย
<	น้อยกว่า
< =	น้อยกว่าหรือเท่ากับ
>	มากกว่า
> =	มากกว่าหรือเท่ากับ
=	เท่ากับ
≠	ไม่เท่ากับ

ค) ตัวปฏิบัติการสำหรับงานทั่วไป ในไมโครซอฟต์แอคเซสได้แบ่งตัวปฏิบัติการชนิดนี้ โดยสรุปดังต่อไปนี้

ตารางที่ 2.4 ตัวปฏิบัติการสำหรับงานทั่วไป

ตัวปฏิบัติการ	ความหมาย
Between ... And	ค่าที่อยู่ในช่วงนี้
In	ค่าที่อยู่ในชุดของข้อมูล
Like	ข้อความที่ตรงกับที่กำหนดไว้แล้วสามารถใช้ร่วมกับอักขร Wildcard แบบ ? และ “
Is	การเปรียบเทียบให้ผลเป็นจริง (เช่น Is Null)
Is Not	การเปรียบเทียบให้ผลเป็นเท็จ(เช่น Is Not Null)
!	นำหน้า Object Name
.	นำหน้าชื่อสำหรับงานหรือคุณสมบัติใดๆ

ง) ตัวปฏิบัติการแบบตรรกศาสตร์ ตัวปฏิบัติการแบบตรรกศาสตร์นี้เป็นการเปรียบเทียบเพื่อสร้างการเปรียบเทียบแบบมีเงื่อนไข ตั้งแต่หนึ่งเงื่อนไขเป็นต้นไป

ตารางที่ 2.5 ตัวปฏิบัติการแบบตรรกศาสตร์

ตัวปฏิบัติการ	ความหมาย
And	เมื่อเปรียบเทียบทั้งสองเป็นจริง
Or	เมื่อเปรียบเทียบตัวใดตัวหนึ่งเป็นจริง
Xor	เมื่อเปรียบเทียบตัวใดตัวหนึ่งเป็นจริง
Nor	เมื่อเปรียบเทียบมีค่าไม่เป็นจริง
Eqv	การเปรียบเทียบแบบ Bitwise ของตัวเลขทั้งสอง
Imp	การเปรียบเทียบแบบ Bitwise ของตัวเลขทั้งสอง

2.12.5.3 ค่าคงที่ และ ข้อมูลเจาะจง (Literal Value) ในการกำหนดค่าคือ การเจาะจงข้อมูลบางกรณี

ก) ตัวเลข เมื่อกำหนดเจาะจงเป็นตัวเลขจะต้องไม่มีเครื่องหมายเว้นวรรคตอนล้อมรอบ ยกตัวอย่างเช่น ในการกำหนดค่า Qty เป็นชื่อฟิลด์ $> =$ เป็นตัวปฏิบัติการและตัวเลข 100 เป็นข้อมูลเจาะจง (Literal Value)

Literal Value [Qty] $> =$ 100

สังเกตว่าตัวเลขนั้นจะไม่ถูกล้อมรอบด้วยเครื่องหมายคำพูด

ข) ข้อความ เมื่อกำหนดค่าข้อความจะต้องกำหนดค่าล้อมรอบด้วยเครื่องหมายคำพูด ยกตัวอย่างการกำหนดค่าต่อไปนี้

[Last Name] = "Jones"

ชื่อ Jones คือ ข้อมูลเจาะจง (Literal Value)

ค) วันที่และเวลา จะต้องล้อมรอบด้วยเครื่องหมาย # ในการกำหนดค่า ถ้าใส่ค่าในไมโครซอฟท์แอคเซส จะใส่ให้แทน ตัวอย่างเช่น การกำหนดค่าของวันที่ตั้งแต่วันที่ 1 กุมภาพันธ์ และวันที่ 15 กุมภาพันธ์ 1999

Between # 2 / 1 / 99 # And # 2 / 15 / 99 #

ง) ค่าคงที่ ค่าคงที่นี้คล้ายกับข้อมูลเจาะจง (Literal Value) แต่ในไมโครซอฟต์เอกเซลนั้นจะกำหนดค่าคงที่ไว้ 5 อย่างด้วยกันคือ Null, Yes, No, True และ False โดยที่ไม่ต้องใช้เครื่องหมายคำพูดหรือเครื่องหมายใดๆล้อมรอบ เช่น ในช่อง Criteria กิวรีได้กำหนดค่า

Is Null เพื่อแยกเรคอร์ดที่ไม่มีบนข้อมูลฟิลด์

Is Not Null เพื่อแยกเรคอร์ดที่มีข้อมูลฟิลด์

Yes ในช่อง Criteria ของฟิลด์ชนิด Yes/No ใน QBE เพื่อแยกเรคอร์ดที่มีค่า Yes (หรือ True หรือ -1) ในฟิลด์นั้นเมื่อใช้ค่าคงที่ในการกำหนดในช่อง Criteria ของคิวรีรวมชื่อฟิลด์ที่คุณต้องการหา ดังตัวอย่างการกำหนดค่าดังนี้

[Country] Is Null

เพื่อแยกเรคอร์ดที่ไม่มีค่าใดๆในฟิลด์ Country ในกรณีที่ต้องการฟิลด์ชนิด Yes/No ในการกำหนด โดยที่ไม่จำเป็นต้องกำหนดค่า Yes และ No ยกตัวอย่างเช่นการกำหนดค่า [Posted] ซึ่งหมายความว่าเช่นเดียวกับ[Posted] = Yes หรือ Posted = True Not[Posted] นั้นมีค่าเท่ากับ[Posted] = No หรือ Posted = False

2.12.5.3 ชื่ออ้างอิง ชื่ออ้างอิงเป็นชื่อที่ถูกใช้ ในนิพจน์เพื่อ ใช้เป็นที่เก็บสำหรับค่าฟิลด์คอนโทรลหรือตัวแปรในหน่วยความจำในแอคเซสเบสิก (Access Basic) ชื่ออ้างอิงนี้ตรงข้ามกับตัวอักษร ชื่ออ้างอิงช่วยสร้างนิพจน์ซึ่งในข้อมูลที่เกี่ยวข้องไว้ในตาราง แบบฟอร์ม และรายงาน ในนิพจน์ต่อไปนี้ ชื่อ (Sales Tax) และ (Sale Price) เป็นชื่ออ้างอิงซึ่งอ้างอิงถึงฟิลด์หรือคอนโทรลบางตัว

$$(\text{Sales Tax}) = 1.065 * (\text{Sales Price})$$

ชื่ออ้างอิงมักจะอยู่ภายในเครื่องหมายวงเล็บเสมอ ดังที่แสดงไว้ในตัวอย่าง อย่างไรก็ตาม ถ้าชื่อ อ้างอิงนั้นเป็นคำคำเดียวเช่น Tax แทนที่จะเป็น วลี เช่น Sales Tax คุณก็สามารถที่จะไม่ใช้วงเล็บในนิพจน์ต่อไปนี้ แต่ ไมโครซอฟท์เอกเซล ก็สามารถที่จะรู้จักกับชื่อ อ้างอิง Tax และ Price ได้

$$(\text{Tax}) = 1.065 * (\text{Price})$$

2.12.5.4 ฟังก์ชัน ฟังก์ชันที่ใช้งานในการคำนวณข้อมูลของไมโครซอฟท์เอกเซลและให้ผลลัพธ์กลับไปยังส่วนควบคุมนั้นมีประมาณ 100 ฟังก์ชัน

ก) ฟังก์ชันช่วย (Help) ในส่วนนี้คุณสามารถใช้ฟังก์ชันช่วย (Help) เพื่อทำการค้นหาฟังก์ชันและข้อมูลการใช้ฟังก์ชันหรือตัวอย่างการใช้ฟังก์ชันต่างๆ

ข) ฟังก์ชันวันที่ / เวลา สรุปฟังก์ชันการใช้งานวันที่ / เวลาดังต่อไปนี้

ตารางที่ 2.6 ฟังก์ชันวันที่ และเวลา

การคำนวณ	ชื่อฟังก์ชัน
วันที่ในปัจจุบัน	Date
วันที่และเวลาในปัจจุบัน	Now
วันที่และเวลาของวันที่/เวลา	Format, Format\$
จำนวนวันระหว่างวันที่	Date Diff
ช่วงเวลา(วัน,สัปดาห์,เป็นต้น)	Date Part
อนุกรมของวันที่	Date Serial

ค) ฟังก์ชันทางด้านข้อความ ฟังก์ชันที่ทำหน้าที่เกี่ยวกับการคำนวณบนข้อความ ซึ่งสรุปได้ดังต่อไปนี้

ตารางที่ 2.7 ฟังก์ชันทางด้านข้อความ

การคำนวณ	ชื่อฟังก์ชัน
การEmbed ตัวหนังสือ	Mid
ตัวอักษรทางด้านท้าย	Right
ตัวอักษรทางด้านหน้า	Left
ความยาว	Len
การเปลี่ยนเป็นตัวเล็ก	Lcase
Null	Is Null
การตัดช่องว่างด้านท้าย	Trim
การตัดช่องว่างด้านหน้า	Ltrim
การเปลี่ยนเป็นตัวใหญ่	Ucase

ง) ฟังก์ชันสรุปผล (Aggregate Function) ฟังก์ชันสรุปผลนี้จะทำการคำนวณทางด้านสถิติทางด้านสถิติบนข้อมูลในตารางหรือคิวรีมีดังนี้

ตารางที่ 2.8 ฟังก์ชันสรุปผล

การคำนวณ	ฟังก์ชัน
หาค่าเฉลี่ย	Avg

นับจำนวน	Count
เรคอร์ดแรกหรือเรคอร์ดสุดท้ายในตาราง หรือ Dynaser	First , Last
ค่าสูงสุด,ค่าต่ำสุด	StDev , StDevP
ผลรวม	Sum
ค่าแปรผัน	Var , VarP

สิ่งสำคัญที่ควรจดจำไว้คือว่าฟังก์ชันเหล่านี้ใช้การคำนวณเฉพาะฟิลด์ในตารางหรือคิวรีเท่านั้น ไม่สามารถที่จะอ้างอิงชื่อของส่วนควบคุมบนฟอร์มหรือรายงานได้ สำหรับรายละเอียดเพิ่มเติมให้ดูที่ฟังก์ชันช่วย (Help) ของฟังก์ชันสรุปผล (Aggregate Functions)

จ) ฟังก์ชันโดเมนสรุปผล (Domain Aggregate) เหมือนกับฟังก์ชันสรุปผล แต่ฟังก์ชันนี้ใช้คำนวณค่าในตารางต่างๆในระบบฐานข้อมูลในปัจจุบันได้ รวมทั้งตารางที่ไม่ได้ถูกเปิดใช้งานสามารถที่จะคำนวณทุกเรคอร์ด หรือเพียงบางส่วนได้

ตารางที่ 2.9 ฟังก์ชันโดเมนสรุปผล (Domain Aggregate)

การคำนวณ	ฟังก์ชัน
ค่าเฉลี่ย	Davg
นับจำนวน	Dcount
เรคอร์ดแรก / สุดท้าย	Dfirst , Dlast
ค่ามากที่สุด / น้อยสุด	Dmax , Dmin
ค้นหา	Dsum
ส่วนเบี่ยงเบนมาตรฐาน	Dlookup
สรุป	DstDev , DvarP
ค่าผันแปร	Dvar , DvarP

2.13 ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ ถือว่าเป็นระบบคอมพิวเตอร์ที่สมบูรณ์ ที่ถูกรวมอยู่ในไอซี (IC : Integrated Circuit) เพียงตัวเดียวโดยรวมวงจรทั้งหมดที่ระบบไมโครโปรเซสเซอร์ (Microprocessor) ต้องใช้ลงในชิปตัวเดียว เรียกว่า ไมโครโปรเซสเซอร์ชิปเดียว บางครั้งเรียก

ว่า ไมโครคอมพิวเตอร์ (Microcomputer) แต่โดยทั่วไปนิยมเรียกว่า ไมโครคอนโทรลเลอร์ซึ่งมีคุณลักษณะที่สำคัญทั่วไปคล้ายกับคอมพิวเตอร์แต่มีขนาดและความสามารถบางอย่างน้อยกว่าและเรายังสามารถต่อเพิ่มได้ตามความต้องการของเราซึ่งเราจะได้ไมโครคอนโทรลเลอร์ เรียกอีกอย่างว่า Computer on Chip

2.13.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

2.13.1.1 ต้องการแหล่งจ่ายไฟ +5 V ชุดเดียว

2.13.1.2 หน่วยประมวลผลทางคณิตศาสตร์ (Arithmetic Logic Unit : ALU) ขนาด 8 บิต

2.13.1.3 หน่วยความจำโปรแกรมภายใน (Read Only Memory : ROM) ขนาด 4 กิโลไบต์

2.13.1.4 หน่วยความจำข้อมูลภายใน (Read Access Memory : RAM) ขนาด 128 ไบต์

2.13.1.5 เคาน์เตอร์ / ไทม์เมอร์ (Counter / Timer) ขนาด 16 บิต 2 ตัว

2.13.1.6 ส่งสัญญาณอินพุต / เอาท์พุท 32 สัญญาณ (พอร์ตอินพุต / เอาท์พุท ขนาด 8 บิต 4 พอร์ต)

2.13.1.7 หน่วยความจำโปรแกรมอแอดเดรส (Program Memory Address Space) ขนาด 64 กิโลไบต์

2.13.1.8 หน่วยความจำข้อมูลอแอดเดรส (Data Memory Address Space) ขนาด 64 กิโลไบต์

2.13.1.9 มีพอร์ตรับส่งข้อมูลอนุกรม 2 พอร์ต ทั้งรับและส่งในเวลาเดียวกันได้ (Full duplex) เลือกรูปแบบการส่งข้อมูลได้ 4 โหมด

2.13.1.10 สัญญาณนาฬิกาความถี่สูงสุด 12 เมกะเฮิรตซ์

2.13.1.11 รับสัญญาณอินเทอร์รัพท์ (Interrupt) ภายนอก 2 สัญญาณ

2.13.1.12 มีคำสั่งในการทำ AD , OR หรือ Complement ได้ทั้งแบบ 8 บิต และ 1 บิต

2.13.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ 8051

MCS-51 ใช้เทคโนโลยีในการผลิตเป็นแบบ NMOS และ CMOS มีโครงสร้างภายในแสดงดังรูปที่ 2.8

รูปที่ 2.8 โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ 8051

2.13.3 โครงสร้างภายนอกของไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 จะมีตำแหน่งขาพื้นฐาน คือ 40 ขาดังแสดงดังรูปที่ 2.9

รูปที่ 2.9 การจัดวางขาของไมโครคอนโทรลเลอร์ตระกูล MCS-51

2.13.3.1 Vcc (ขา 40) ต่อกับไฟเลี้ยง +5 V

2.13.3.2 Vss (ขา 20) เป็นขากราวน์ (GND)

2.13.3.3 พอร์ต 0 (ขา 32- 39) มีทั้งหมด จำนวน 8 บิต P 0.0 - P 0.7 ใช้งานได้ 2 หน้าที คือ ส่งแอดเดรส (Adress) และ ข้อมูล (Data) ออกไปให้หน่วยความจำภายนอก เมื่อทำการเขียน ข้อมูลลงในหน่วยความจำภายนอกควบคุมด้วยขาคอนโทรล ดังรูปที่ 2.10 และอีกหน้าที่หนึ่งก็คือ เป็นพอร์ตอินพุท/เอาต์พุท (I/O Port) ถ้าต้องการให้ทำงานส่วนที่เป็นพอร์ตอินพุทต้องเขียน โปรแกรมส่งลอจิก 1 ไปยังพอร์ตนี้

2.13.3.4 พอร์ต 1 (ขา 1-8) มีทั้งหมด 8 บิต คือ P1.0 – P1.7 มีโครงสร้างคล้ายพอร์ต 0 แต่ จะใช้ความต้านทานภายในพูลอัพแทน (Internal Pull Up Register) มีโครงสร้างภายในดังรูปที่ 2.11

2.13.3.5 พอร์ต 2 (ขา 21- 28) มีทั้งหมด 8 บิต คือ P2.0 – P2.7 มีโครงสร้างคล้ายพอร์ต 0 โดยมี FET ตัวเดียวอีกส่วนหนึ่งใช้ความต้านทานพูลอัพแทน พอร์ตนี้ทำงาน 2 หน้าทีคือ

สามารถใช้เป็นพอร์ตสำหรับส่งแอดเดรส 8 บิตบน และเป็นพอร์ตอินพุท/เอาต์พุท (I/O Port) ใช้งานทั่วไป เมื่อจะใช้งานเป็นอินพุทพอร์ต ต้องส่งลอจิก 1 มาที่พอร์ตนี้

รูปที่ 2.10 แสดงโครงสร้างของพอร์ต 0

รูปที่ 2.11 แสดงโครงสร้างของพอร์ต 1

รูปที่ 2.12 แสดงโครงสร้างของพอร์ต 2

2.13.3.6 พอร์ต 3 (ขา10-17) มีทั้งหมด 8 บิต คือ P 3.0 - P 3.7 มีโครงสร้างคล้ายพอร์ต 1 พอร์ตนี้ทำหน้าที่คือเป็นพอร์ตอินพุท/เอาต์พุท (I/O Port) ถ้าจะให้พอร์ตนี้เป็นอินพุทพอร์ต ก็ให้ส่งลอจิก 1 มาที่พอร์ตนี้ก่อน และอีกหน้าที่หนึ่งก็คือ ส่งสัญญาณควบคุมออกมา และรับสัญญาณเข้าไป สัญญาณต่างๆดังต่อไปนี้

- ก) P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับสัญญาณแบบอนุกรม
- ข) P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งสัญญาณแบบอนุกรม
- ค) P3.2/INT0 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก
- ง) P3.3/INT1 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก
- จ) P3.4/T0 (Timer/Counter 0 External Input) ขารับสัญญาณเข้าไปยังวงจรถ่ายเวลา/เคาน์เตอร์ 0 (Timer/Counter 0) ที่ทำหน้าที่นับจำนวนไซเคิล (Cycle) ของเวลาที่ใช้ประมวลผลคำสั่ง หรือสัญญาณนาฬิกาได้

ฉ) P3.5/T1 (Timer/Counter 1 External Input) ขารับสัญญาณเข้าไปยังวงจรถ่ายเก็บ/เลขคณิตเตอร์ 1 (Timer/Counter1) ที่ทำหน้าที่นับจำนวนไซเคิล (Cycle) ของเวลาที่ใช้ประมวลผลคำสั่ง หรือสัญญาณนาฬิกาได้ ซึ่งมีการทำงานเหมือน T0

ช) P3.6/WR (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051

ซ) P3.7/RD (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

รูปที่ 2.13 แสดงโครงสร้างของพอร์ต 3

ฅ) ALU (ขา 30) เป็นขาที่ใช้ส่งสโตรบ (Strobe) สำหรับใช้ในการแลทช์ (Latch) แอดเดรส (Address) ไบต์ต่ำที่ส่งออกมาจากพอร์ต 0 สัญญาณนี้จะแอคทีฟ (Active) ทุกๆ 2 ครั้ง ใน 1 แมชชีนไซเคิล (Machine Cycle) หรือ $\frac{1}{16}$ ของสัญญาณนาฬิกา

ญ) PSEN (ขา 29) เป็นขาที่ใช้ส่งสโตรบ (Strobe) สำหรับใช้ในการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก (External Program Memory) หน่วยความจำประเภท ROM หรือ EPROM สัญญาณนี้จะส่งออกมา 2 ครั้ง ในแต่ละแมชชีนไซเคิล (Machine Cycle) แต่ถ้าเป็นการอ่านหน่วยความจำโปรแกรมภายใน (Internal Program Memory) จะไม่มี สัญญาณออกที่ขานี้

ฎ) EA (ขา 31) ถ้าป้อนลอจิก 0 เข้าที่ขานี้หน่วยประมวลผลกลาง (Central Processing Unit : CPU) จะอ่านค่าจากหน่วยความจำโปรแกรมภายนอก (External Program Memory) แต่ถ้าถูกป้อนด้วยลอจิก 1 ก็ จะอ่านค่าจากหน่วยความจำโปรแกรมภายใน (Internal Program Memory)

ฏ) RST (ขา 9) เป็นขารีเซ็ต (Reset) หน่วยประมวลผลกลาง (CPU) เมื่อป้อนลอจิก 1 เข้าที่ขานี้ นานอย่างน้อย 2 แมชชีนไซเคิล (Machine Cycle)

ฐ) XTAL1 (ขา 19) ใช้ต่อคริสตอล (Crystal) ภายนอกโดยเป็นอินพุต (Input) เข้าสู่วงจรรอสซิเลเตอร์ (Oscillator Circuit)

ฑ) XTAL 2 (ขา 18) ใช้ต่อคริสตอล (Crystal) ภายนอกโดยเป็นอินพุต (Input) เข้าสู่วงจรรอสซิเลเตอร์ (Oscillator Circuit)

2.13.4 การแบ่งประเภทของหน่วยความจำ

ในระบบ MCS-51 จะมีการแบ่งหน่วยความจำเหมือนกับหน่วยประมวลผลกลาง (CPU) ทั่วๆไปจะแบ่งเป็น 2 ลักษณะตามชนิดของข้อมูลที่เก็บดังนี้

2.13.4.1 หน่วยความจำโปรแกรม (Program Memory)

หน่วยความจำโปรแกรม ใน MCS-51 จะแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำโปรแกรมภายในและหน่วยความจำโปรแกรมภายนอก หน่วยความจำภายในก็คือหน่วยความจำประเภท รอม หรืออีพรอม (Read Only Memory : ROM หรือ Erasable Programmable Read Only Memory : EPROM) ที่อยู่ภายในตัว MCS-51 ซึ่งมีขนาด 4 กิโลไบต์ ส่วนหน่วยความจำโปรแกรมภายนอกก็คือ หน่วยความจำต่ออยู่ภายนอกตัว MCS-51 โดยอาจจะต่อเพื่อขยายหน่วยความจำเมื่อหน่วยความจำไม่พอ โดย MCS-51 สามารถเลือกให้รันโปรแกรมในหน่วยความจำภายในและภายนอกได้ ดังรูปที่ 2.14

รูปที่ 2.14 แผงหน่วยความจำโปรแกรม (Program Memory) ของไมโครคอนโทรลเลอร์เบอร์ 8051

2.13.4.2 หน่วยความจำข้อมูล (Data Memory)

หน่วยความจำข้อมูล เป็นหน่วยความจำที่สามารถอ่านหรือเขียนข้อมูลเปลี่ยนแปลงได้ตลอดเวลา แต่ไม่สามารถรัน (Run) โปรแกรมบนหน่วยความจำส่วนนี้ได้ หรือ เรียกหน่วยความจำข้อมูล (Read Access Memory : RAM) ซึ่งเป็นหน่วยความจำภายในชิพ (Chip) มีเพียง 128 ไบต์ และยังมีส่วนหน่วยความจำภายนอกมี 64 กิโลไบต์ ดังรูปที่ 2.15

ตำแหน่งของหน่วยความจำสำหรับโปรแกรมและข้อมูลที่มีตำแหน่งซ้อนกันแยกคำสั่งออกเป็น 3 ส่วน คือ

คำสั่ง MOV ใช้ติดต่อกับหน่วยความจำข้อมูลภายใน (Internal RAM)

คำสั่ง MOVC ใช้ติดต่อกับหน่วยความจำโปรแกรมภายนอก (External ROM)

คำสั่ง MOVX ใช้ติดต่อกับหน่วยความจำข้อมูลภายนอก (External RAM) โดยระบุตำแหน่งผ่านรีจิสเตอร์ดีพิตีอาร์ (Data Pointer Register : DPTR) สำหรับหน่วยความจำข้อมูลภายในนั้นจะแบ่งออกได้ดังนี้

ก) ชุดรีจิสเตอร์ 4 ชุดแต่ละชุดเรียกว่ารีจิสเตอร์แบงก์ (Default Register Bank) ที่ตำแหน่ง 00H ถึง 1FH โดยแต่ละชุดประกอบด้วยรีจิสเตอร์ R0 ถึง R7

- ข) หน่วยความจำที่สามารถเข้าถึงข้อมูลระดับบิตได้ (Bit-address Location) ตำแหน่ง 20H ถึง 2FH
- ค) หน่วยความจำใช้งานทั่วไป (General Purpose RAM) ตำแหน่ง 30H ถึง 7FH
- ง) รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register) ตำแหน่ง 80H ถึง FFH

รูปที่ 2.15 แผงหน่วยความจำข้อมูล (Data Memory) ของไมโครคอนโทรลเลอร์เบอร์ 8051

2.14 ชุดคำสั่งของ MCS-51

ชุดคำสั่งของ 8051 มีทั้งหมด 111 คำสั่ง โดยแบ่งออกเป็นคำสั่งขนาดไบต์เดียว 49 คำสั่ง คำสั่งขนาด 2 ไบต์ 45 คำสั่ง และคำสั่งขนาด 3 ไบต์ 17 คำสั่ง แต่ในที่นี้จะขอล่าวถึงเฉพาะคำสั่งที่ใช้ในการทำโครงการนี้เท่านั้น วิธีการเปลี่ยนแปลงภาษาแอสเซมบลีให้เป็นภาษาเครื่องสามารถดำเนินการได้โดยใช้โปรแกรมแอสเซมเบลอร์ (Assembler) เช่น SXA-51 , XTALK , CROSS-32 โดยจะต้องเขียน โปรแกรมที่ใช้พิมพ์เก็บไว้ในไฟล์ (File) โดยตั้งชื่อให้มีนามสกุลเป็น .ASM ตัวอย่างของคำสั่งมีดังนี้

2.14.1 คำสั่งโอนย้ายข้อมูล

2.14.1.1 การโอนย้ายข้อมูลกับหน่วยความจำข้อมูลภายใน

ก) การนำข้อมูลมาเก็บไว้ในรีจิสเตอร์ A มีรูปแบบและความหมายดังนี้

รูปแบบคำสั่ง	ความหมาย
MOV A , Rn	นำข้อมูลจากรีจิสเตอร์ R0 – R7 มาเก็บไว้ยังรีจิสเตอร์ A
MOV A , Direct	นำค่าข้อมูลจากหน่วยความจำข้อมูลภายใน มาเก็บยังรีจิสเตอร์ A
MOV A , # Data	นำค่าข้อมูลตัวเลขมาเก็บไว้ยังรีจิสเตอร์ A

ข) การนำข้อมูลมาเก็บไว้ในรีจิสเตอร์ทั่วไป มีรูปแบบคำสั่งและความหมายดังนี้

รูปแบบคำสั่ง	ความหมาย
MOV Rn , A	นำค่าข้อมูลจากรีจิสเตอร์ A มาเก็บไว้ยัง รีจิสเตอร์ Rn
MOV Rn , Direct	นำค่าข้อมูลจากหน่วยความจำข้อมูลภายในมาเก็บไว้ยังรีจิสเตอร์ Rn
MOV Rn , # Data	นำค่าข้อมูลตัวเลขมาเก็บไว้ยังรีจิสเตอร์ Rn

ค) การนำข้อมูลมาเก็บไว้ยังหน่วยความจำโดยตรง มีรูปแบบคำสั่งและความหมายดังนี้

รูปแบบคำสั่ง	ความหมาย
MOV Direct , # Data	นำค่าตัวเลขมาเก็บไว้ยังหน่วยความจำภายใน

2.14.1.2 การโอนย้ายข้อมูลกับหน่วยความจำข้อมูลภายนอก

รูปแบบคำสั่ง	ความหมาย
MOVX A , @Rn	นำข้อมูลจากหน่วยความจำที่มีค่าเก็บอยู่ภายในรีจิสเตอร์ Rn มาเก็บยังรีจิสเตอร์ A
MOVX A , @DPTR	นำข้อมูลจากหน่วยความจำที่มีค่าเก็บอยู่ภายในรีจิสเตอร์ DPTR มาเก็บไว้ยังรีจิสเตอร์ A
MOVX @DPTR , A	นำข้อมูลจากรีจิสเตอร์ A มาเก็บไว้ยังหน่วยความจำที่มีค่าเก็บอยู่ภายในรีจิสเตอร์ DPTR

2.14.1.3 การโอนข้อมูลจากหน่วยความจำโปรแกรมภายนอก การโอนย้ายข้อมูลจะเป็นเฉพาะการอ่านข้อมูลจากหน่วยความจำโปรแกรมแต่เพียงอย่างเดียวเท่านั้น การนำข้อมูลจากหน่วยความจำของโปรแกรมโดยการอ้อมผ่านทางรีจิสเตอร์ มีรูปแบบคำสั่งดังนี้

รูปแบบคำสั่ง	ความหมาย
MOVC A , @A+DPTR	นำข้อมูลจากหน่วยความจำโปรแกรม โดยทำการพิจารณาตำแหน่งจากการบวกค่ารีจิสเตอร์ A กับค่าของรีจิสเตอร์ DPTR
MOVC A , @A+PC	นำข้อมูลจากหน่วยความจำโปรแกรม โดยพิจารณาตำแหน่งจากการบวกค่ารีจิสเตอร์ A กับค่าของรีจิสเตอร์ PC

2.14.2 คำสั่งการคำนวณทางคณิตศาสตร์

รูปแบบคำสั่ง	ความหมาย
INC A	ทำการบวกหนึ่ง กับค่าข้อมูลภายในรีจิสเตอร์ A
DEC A	ทำการลบหนึ่งจากค่าข้อมูลภายในรีจิสเตอร์ A

2.14.3 กลุ่มคำสั่งทางตรรกศาสตร์

การใช้คำสั่งนี้การทำงานเหมือนกับพีชคณิตของบูลีน (Boolean Operation) ซึ่งได้แก่ AND, OR, XOR, NOT มีประโยชน์ในการใช้งานควบคุมสวิตช์ต่างๆรูปแบบคำสั่งของ MCS-51 เมื่อเทียบกับ พีชคณิตของบูลีน มีดังนี้

พีชคณิตของบูลีน (Boolean Operation)	รหัสนีโมนิก (Mnemonic Code) ของ MCS-51
AND	ANL (AND Logical)
OR	ORL (OR Logical)
XOR	XRL (Exclusive OR Logical)
NOT	CPL (Complement)

การกระทำทางลอจิกสามารถดำเนินการได้ ทั้งบิตและไบต์ มีรายละเอียดดังนี้

2.14.3.1 การดำเนินการทางตรรกกับข้อมูลแบบไบต์ มีรูปแบบคำสั่งและความหมายดังนี้

รูปแบบคำสั่ง	ความหมาย
ANL A , #Data	ทำการ AND แต่ละบิตข้อมูลของรีจิสเตอร์ A กับบิตภายในของตัวเลข Data ที่มีตำแหน่งตรงกัน และเก็บค่าผลลัพธ์ไว้ในรีจิสเตอร์ A
ANL Direct , #Data	ทำการ AND แต่ละบิตข้อมูลของหน่วยความจำแอดเดรส Direct กับแต่ละบิตภายในตัวเลข Data ที่มีตำแหน่งตรงกันและเก็บค่าของผลลัพธ์ไว้ยังหน่วยความจำ Direct
ORL A , #Data	ทำการ OR แต่ละบิตข้อมูลของรีจิสเตอร์ A แต่ละบิตภายใน Data ตรงกันและเก็บค่าผลลัพธ์ไว้ในรีจิสเตอร์ A
CLR A	กำหนดค่าของแต่ละบิตภายในรีจิสเตอร์ A เป็น 0

2.14.3.2 การดำเนินการทางตรรกกับข้อมูลแบบบิต มีรูปแบบคำสั่งและความหมายดังนี้

รูปแบบคำสั่ง	ความหมาย
SETB Bit	ทำการเซตค่าบิตของข้อมูลในตำแหน่ง บิตแอดเดรส Bit เป็น 1

2.14.4 คำสั่งกระโดด (Jump)

2.14.4.1 คำสั่งกระโดดแบบไม่มีเงื่อนไข (Unconditional Jump) มีรูปแบบคำสั่งและความหมายดังนี้

รูปแบบคำสั่ง	ความหมาย
JMP @A+DPTR	ทำการกระโดดไปยังหน่วยความจำโปรแกรมโดยพิจารณาจากตำแหน่งจากการบวกค่ารีจิสเตอร์ A กับค่าของรีจิสเตอร์ DPTR
AJMP Addr11	กระโดดไปยังตำแหน่ง Addr11 โดยใช้ค่าของตัวเลขสัมบูรณ์เพียง 11 บิต และไม่มีเงื่อนไขในการกระโดด
LJMP Addr16	กระโดดไปยังตำแหน่ง Addr16 โดยใช้ค่าของตัวเลขสัมบูรณ์เพียง 16 บิต และไม่มีเงื่อนไขในการกระโดด

2.14.4.2 คำสั่งกระโดดแบบมีเงื่อนไข (Conditional Jump) มีรูปแบบคำสั่งและความหมายดังนี้

รูปแบบคำสั่ง	ความหมาย
JB Bit , Rel	กระโดดไปยังหน่วยความจำโปรแกรมตำแหน่งสัมพัทธ์ Rel เมื่อค่าของบิตที่ตำแหน่ง Bit ภายในหน่วยความจำภายในเป็น 1
JNB Bit , Rel	กระโดดไปยังหน่วยความจำโปรแกรมตำแหน่งสัมพัทธ์ Rel เมื่อค่าของบิตที่ตำแหน่ง Bit ภายในหน่วยความจำภายในเป็น 1

CJNE Rn , #Data , Rel	ทำการเปรียบเทียบค่าในรีจิสเตอร์ R0-R7 กับ ค่าของตัวเลข Data เมื่อมีค่าไม่เท่ากันจึงกระโดดไปยังหน่วยความจำโปรแกรมตำแหน่งสัมพันธ์ Rel
DJNZ Direct , Rel	ทำการลดค่าภายใน หน่วยความจำตำแหน่งที่มีค่าเก็บอยู่ภายในรีจิสเตอร์ R0 หรือ R1 จากนั้นจึงพิจารณาว่าหากค่าในรีจิสเตอร์ R0-R7 นั้นไม่เป็นค่าศูนย์แล้ว ก็จะต้องกระโดดไปยังหน่วยความจำโปรแกรมตำแหน่งสัมพันธ์ Rel

2.14.5 คำสั่งเกี่ยวกับการเรียกและการกลับจากโปรแกรมย่อย

รูปแบบคำสั่ง	ความหมาย
ACALL Addr11	ทำการเรียกโปรแกรมย่อย ซึ่งอยู่ในตำแหน่ง Addr11 โดยใช้ค่าของตัวเลขสัมพันธ์ไม่เกิน 11 บิต
LCALL Addr16	ทำการเรียกโปรแกรมย่อยที่อยู่ในตำแหน่งแอดเดรสแห่งใดก็ได้ที่อยู่ในขอบเขตการอ้างอิงของหน่วยความจำที่สามารถทำงานได้
RET	ทำการนำค่า (Pop) ข้อมูล 2 ไบต์ ออกมาจากบริเวณสแต็กพอยน์เตอร์ (Stack Pointer) และไปใช้กับรีจิสเตอร์ PC (Program Counter Register) หรืออาจกล่าวได้ว่าเป็นการสิ้นสุดการทำงานของซับรูทีน (Sub-Routine)

2.15 มอเตอร์ไฟฟ้ากระแสตรง (Direct Current Motor)

หลักการของมอเตอร์ก็คือการเปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานกล เพื่อนำพลังงานกลที่ได้ไปขับเคลื่อนสิ่งต่างๆตามที่ต้องการ โครงสร้างส่วนต่างๆของมอเตอร์ก็เหมือนกับโครงสร้างของเครื่องกำเนิดไฟฟ้าสามารถใช้แทนกันได้ส่วนการทำงานของมอเตอร์สามารถอธิบายได้ดังนี้

รูปที่ 2.16 หลักการทำงานของมอเตอร์

จากรูปที่ 2.16 แสดงส่วนหนึ่งของมอเตอร์ไฟฟ้ากระแสตรงที่มีหลายขั้วแม่เหล็ก เมื่อใส่ไฟเข้าไปที่สนามกระตุ้นยอมทำให้เกิดอำนาจแม่เหล็กขึ้นที่ขั้วแม่เหล็ก และเมื่อป้อนกระแสไฟให้ไหลผ่านในตัวนำที่อาร์เมเจอร์ (Armature) ก็จะทำให้เกิดแรงขึ้นในตัวนำที่อยู่ในสนามแม่เหล็ก นั่นคือทำให้ อาร์เมเจอร์หมุน ในที่นี้ได้กำหนดให้กระแสไหลในตัวนำของอาร์เมเจอร์ที่อยู่ภายใต้ขั้วเหนือ N มีทิศทางของกระแสพุ่งเข้าไปข้างในดังหางลูกศรที่เป็นกากบาท ส่วนตัวนำที่อยู่ภายใต้ขั้ว S นั้นให้กระแสพุ่งออกมาข้างนอกดังหัวลูกศรที่เป็นจุด จะพบว่าแรงที่เกิดขึ้นบนตัวนำทุกๆตัว ภายใต้ขั้วแม่เหล็กเดียวกัน จะมีทิศทางไปในทางเดียวกัน และจะอยู่สัมผัสกับเส้นรอบวงของอาร์เมเจอร์นั้นๆ และพบว่าแรงที่เกิดขึ้นภายใต้ขั้วแม่เหล็กทุกๆขั้วที่สลับกันไปมานั้นจะมีทิศทางไปในทางเดียวกันทั้งสิ้น โดยแต่ละแรงจะอยู่ในแนวเส้นสัมผัสกับเส้น รอบวงของอาร์เมเจอร์นั้นคือ ภายใต้ขั้วแม่เหล็กแต่ละขั้วก็จะเกิดแรงลัพธ์ขึ้นหนึ่งแรง ฉะนั้นเมื่อมีหลายขั้วก็มีหลายแรงและทุกๆแรงต่างก็รวมกันเป็นแรงบิดขึ้นมาแรงหนึ่ง ดังนั้นจึงทำให้มอเตอร์หมุนไปได้จากแรงที่เกิดขึ้นนี้

2.16 ไดโอด (Diode)

ไดโอดเป็นอุปกรณ์ทางโซลิดสเตท (Solid State) ที่มีขั้ว 2 ขั้ว และยอมให้กระแสไหลผ่านตัวมันทางเดียวเท่านั้น ถ้ากระแสไหลในทิศทางตรงข้ามจะไม่สามารถไหลผ่านตัวมันได้ ดังรูปที่ 2.17 แสดงสัญลักษณ์ของไดโอด และแสดงโครงสร้างของไดโอดพื้นฐานทั่วไป ซึ่งเป็นรอยต่อ P-N โดยขั้ว P เรียกว่า แอโนด (Anode) ขั้ว N เรียกว่า แคโทด (Cathode)

รูปที่ 2.17 สัญลักษณ์และโครงสร้างพื้นฐานของไดโอด

ก)

ข)

รูปที่ 2.18 ก) ไดโอดขณะไบอัสตรง (Forward Bias) ข) ไดโอดขณะไบอัสกลับ (Reverse Bias)

จากรูปที่ 2.18 ก) เป็นการแสดงคุณลักษณะพื้นฐานของไดโอดเมื่อได้รับการไบอัสตรง (แอนโอดมีศักย์ไฟฟ้าเป็นบวกเมื่อเทียบกับแคโทด) ทำให้มีลักษณะ เป็นค่าความต้านทานต่ำๆ กระแสจึงสามารถไหลผ่านไดโอดได้ แต่จากรูปที่ 2.18 ข) เมื่อไดโอดได้รับการไบอัสกลับ (แอนโอดมีศักย์ไฟฟ้าเป็นลบเมื่อเทียบกับแคโทด) ทำให้มีลักษณะเป็นความต้านทานสูงกระแสจึงไม่สามารถไหลผ่านไดโอดได้

2.17 ไดโอดเปล่งแสงกำลังสูงที่ให้แสงอินฟราเรด (High Power Near-Infrared Emitting Diode)

ไดโอดเปล่งแสงกำลังสูง ที่ให้แสงย่านอินฟราเรด (High Power Near - Infrared Emitting Diode) และเป็นแหล่งกำเนิดโฟตอน (Photon) ที่ดีมาก

การเปล่งแสงของไดโอดเปล่งแสงกำลังสูงมีความยาวคลื่น 880 นาโนเมตร จะใกล้เคียงกับความยาวคลื่นที่ซิลิกอนโฟโตทรานซิสเตอร์ (Silicon Photo Transistor) มีความไวสูงสุด จึงเป็นการเหมาะสมที่จะใช้ไดโอดเปล่งแสงชนิดนี้เป็นแหล่งกำเนิดแสงย่านอินฟราเรดเพื่อส่งสัญญาณแสงในการไปอัสซิลิกอนโฟโตทรานซิสเตอร์ (Silicon Photo Transistor)

2.18 ทรานซิสเตอร์ (Transistor)

ทรานซิสเตอร์ (Transistor : Tr หรือ Q) เป็นอุปกรณ์สารกึ่งตัวนำที่สาร P และสาร N 3 ชั้นมาต่อเรียงกันดังรูปที่ 2.19 โดยเรียงต่อกันได้ 2 แบบ ดังรูปที่ 2.19 ก) และ ข) ในรูปที่ 2.19 ก) ใช้สาร N 2 ชั้นและสาร P 1 ชั้น โดยมีสาร P อยู่ตรงกลาง ซึ่งเรียกทรานซิสเตอร์ชนิดนี้ว่า NPN และต่อออกมา 3 ขา เป็นขาเบส (B:Base) ขาคอลเลคเตอร์ (C:Collector) และขาอีมิเตอร์ (E:Emitter) โดยขาเบส ต่อออกจากสาร P ส่วนในรูปที่ 2.19 ข) ตรงข้ามกับรูปที่ 2.19 ก) และเรียกว่าชนิด PNP ส่วนขาที่ต่อออกมาก็เหมือนกับรูปที่ 2.19 ก) ด้วยโครงสร้างดังกล่าวนี้จะเหมือนกับไดโอด 2 ตัวมาต่อชนกันดังรูปที่ 2.19 ค) และ ง) โดยใช้สาร P หรือ N ตรงกลางร่วมกัน

รูปที่ 2.19 โครงสร้างของทรานซิสเตอร์ ก) ชนิด NPN ข) ชนิด PNP

ค) และ ง) การเปรียบเทียบเสมือนไดโอด 2 ตัวชนกัน

รูปที่ 2.20 สัญลักษณ์ของทรานซิสเตอร์ชนิด ก) ชนิดNPN ข) ชนิดPNP

หลักการของทรานซิสเตอร์อธิบายได้โดยการต่อวงจรดังรูปที่ 2.21 ด้วยแบตเตอรี่ และตัวต้านทานในรูปที่ 2.21 ก) เป็นการต่อในลักษณะไบอัสตรงให้กับสาร P และ N ด้วย V_B (เหมือนไดโอด) จึงมีกระแสส่วนหนึ่งไหลได้ชื่อว่า I_{BE} ซึ่งเป็นผลให้ทางขา C เกิดกระแสไหลตามไปด้วยคือ มีกระแสวิ่งจากแบตเตอรี่ V_S ไปสาร N ไปสาร P ที่ E ครบวงจรอีก และกระแส ส่วนที่วิ่งไหลตาม I_{BE} นี้มีชื่อว่า I_{CE} และกระแสที่วิ่งออกจากขา E มี 2 ส่วนคือ ส่วนของ I_{BE} และ E_{CE} ส่วนรูปที่ 2.21 ข) ก็จะมีหลักการทำงานเช่นเดียวกับชนิด PNP แต่เพียงกลับขั้วแบตเตอรี่เท่านั้น และเมื่อหากว่า I_{BE} หยุดไหล I_{CE} จะหยุดไหลตามไปด้วยเช่นกัน

รูปที่ 2.21 การเกิดกระแสเมื่อมีการป้อนแรงดันที่ขาต่างๆ

รูปที่ 2.22 วงจรจากการต่อวงจรในรูปที่ 2.21

2.19 โฟโตทรานซิสเตอร์ (Photo Transistor)

ซิลิกอนทรานซิสเตอร์นั้นจะถูกสร้างจากรอยต่อ P-N ทรานซิสเตอร์จึงมีความไวต่อแสง ซึ่งอยู่ในรูปแบบของโฟโตทรานซิสเตอร์นั่นเองสัญลักษณ์ของโฟโตทรานซิสเตอร์แสดงในรูปที่ 2.21

รูปที่ 2.23 สัญลักษณ์ของโฟโตทรานซิสเตอร์

การใช้งานของโฟโตทรานซิสเตอร์พื้นฐานมีอยู่ 3 แบบด้วยกัน ดังแสดงในรูปที่ 2.25 จากรูปจะเห็นว่าในแต่ละกรณีรอยต่อเบส-คอลเลคเตอร์ ของทรานซิสเตอร์ซึ่งต่อในลักษณะได้รับไบอัส กลับเช่นเดียวกับโฟโตไดโอด ในรูปที่ 2.25 (ก) ขาเบสของทรานซิสเตอร์ถูกต่อลงกราวด์ดังนั้นทรานซิสเตอร์จะทำงานเหมือนกับโฟโตไดโอดทุกประการ ส่วนรูปที่ 2.25 (ข) และ 2.25 (ค) ขาเบสของทรานซิสเตอร์ถูกปล่อยลอยไว้ และ เมื่อใดที่ทรานซิสเตอร์ได้รับแสงก็จะมีกระแสไหลผ่านรอยต่อเบสคอลเลคเตอร์ยังขาเบสของทรานซิสเตอร์ ซึ่งทำให้กระแสที่ผ่านจากคอลเลคเตอร์มายังอิมิตเตอร์ของทรานซิสเตอร์ซึ่งทำให้กระแสที่ผ่านจากคอลเลคเตอร์มายังอิมิต

เตอร์ของทรานซิสเตอร์มีปริมาณเพิ่มขึ้นอย่างมากกระแสนี้จะทำให้ได้แรงดันเอาต์พุตที่ตกคร่อมตัวต้านทาน R1 ที่ต่ออนุกรมอยู่มีค่าเพิ่มขึ้น

รูปที่ 2.24 วงจรโฟโต้ทรานซิสเตอร์ที่สามารถปรับความไวในการตอบสนองต่อแสง

รูปที่ 2.25 การใช้งานของโฟโต้ทรานซิสเตอร์ในลักษณะต่างๆ

2.20 ออปโตคัปเปิลเลอร์ (Opto - Coupler)

ออปโตคัปเปิลเลอร์เป็นอุปกรณ์ที่ประกอบด้วยไดโอดเปล่งแสง (Light Emitting Diode : LED) ซึ่งปกติเป็นชนิดอินฟราเรดและโฟโตทรานซิสเตอร์ (Photo Transistor) ที่ถูกผลิตมาเป็นคู่กัน รวมอยู่ในตัวถังเดียวกันดังรูปที่ 2.26

รูปที่ 2.26 วงจรออปโตคัปเปิลเลอร์ (Opto-Coupler) พื้นฐาน

จากรูปที่ 2.26 เป็นวงจรใช้งานพื้นฐานของออปโตคัปเปิลเลอร์ โดยด้านไดโอดเปล่งแสง(LED)จะเป็นอินพุทของวงจร และด้านโฟโตทรานซิสเตอร์เป็นด้านเอาต์พุทของวงจร จากรูปเมื่อสวิตช์ SW_1 เปิดวงจร ไดโอดเปล่งแสงและทรานซิสเตอร์ Q1 จะยังไม่ทำงานแต่เมื่อ SW_1 ปิดวงจรจะมีกระแสไหลผ่าน ไดโอดเปล่งแสงและตัวต้านทาน R1 ซึ่งจะทำให้ Q1 ทำงานด้วยและจะได้แรงดันเอาต์พุทที่ตัวต้านทาน R2 จะเห็นได้ว่าเอาต์พุทของวงจรถูกควบคุมโดยส่วนอินพุท แต่ทั้งสองแยกจากกันทางไฟฟ้าอย่างสิ้นเชิง (ซึ่งเป็นหลักการสำคัญของออปโตคัปเปิลเลอร์) วงจรนี้สามารถนำไปประยุกต์ใช้ได้กับสัญญาณอนาล็อก (Analog) และสัญญาณดิจิทัล (Digital)

2.21 หลักการระบายความร้อนให้กับอุปกรณ์สารกึ่งตัวนำ

วงจรอิเล็กทรอนิกส์ยังคงต้องใช้กระแสไฟฟ้าในการทำงาน ยิ่งกระแสไฟฟ้ามากขึ้นเท่าใดกำลังงานสูญเสียเกิดขึ้นกับอุปกรณ์ในวงจรทุกๆตัว และกำลังงานที่สูญเสียไปนี้เองจะถูกแปรสภาพออกมาเป็นความร้อน อุปกรณ์ที่เกิดการแปรสภาพ ออกมาเป็นความร้อนมาให้เห็นอย่างชัดเจน ได้แก่ ทรานซิสเตอร์ ตัวต้านทาน เป็นต้น โดยสังเกตได้ง่ายๆที่ตัวถังของอุปกรณ์ ซึ่งจะเป็นตัวถังโลหะถ้าพื้นที่ของตัวถังโลหะมากขึ้นเท่าใดแสดงว่าสามารถเป็นตัวกำเนิดความร้อนได้มากขึ้นเท่า

นั้น แต่ก็ยังไม่สามารถระบายความร้อนออกนอกรอบตัวถังได้เพียงพอ จึงจำเป็นต้องอาศัยสิ่งที่จะมาช่วยระบายความร้อนแล้วถ่ายเทออกไปได้เร็ว และสิ่งนี้ก็คือแผ่นระบายความร้อน (Heat Sink) เพื่อให้เกิดการนำความร้อนได้ดีที่สุด อัตราการไหลของความร้อนต้องมีค่ามากๆ นั่นคือ วัสดุที่ใช้ควรจะมีค่าการนำความร้อนสูงๆ และมีพื้นที่หน้าตัดขวางมากๆ ด้วยเช่นกัน นอกจากนี้อุณหภูมิรอบข้างควรจะต่ำและความหนาของวัสดุควรจะมีค่าน้อย นอกจากนี้ความร้อนยังถูกถ่ายเทด้วยขบวนการพาความร้อนและแผ่รังสีความร้อน เมื่อพื้นผิวของวัสดุร้อนกว่าอากาศรอบข้าง ความหนาแน่นอากาศจะเบาบางลง จึงทำให้อากาศรอบๆ ตัวลอยขึ้นไปพร้อมกับพาเอาความร้อนตามไปด้วย ส่วนการแผ่รังสีความร้อนนั้นจะกระจายออกมาที่รอบๆ ตัวของวัสดุ ซึ่งขึ้นอยู่กับพื้นที่ผิวทั้งหมด อุณหภูมิ และ คุณสมบัติในการแผ่รังสีของวัสดุเองจากทั้งหมดนี้จึงพอสรุปออกมาได้ว่าแผ่นระบายความร้อนที่ดีควรมีคุณสมบัติดังต่อไปนี้

- 2.21.1 มีพื้นที่ผิวมากที่สุด (ด้วยวิธีการทำครีบบระบายความร้อนหลายๆ)
- 2.21.2 ทำด้วยวัสดุที่มีค่าการนำความร้อนสูงๆ (เงิน, ทองแดง, ทองหรืออะลูมิเนียม)
- 2.21.3 มีคุณสมบัติในการแผ่รังสีความร้อนได้ดี (ด้วยวิธีเคลือบสีอะลูมิเนียมหรือทองแดง)
- 2.21.4 ออกแบบให้ติดตั้งมาในตำแหน่งที่เหมาะสมและระบายอากาศได้ดี
- 2.21.5 ออกแบบให้ติดตั้งอุปกรณ์กำลังต่ำอยู่ต่ำกว่าอุปกรณ์กำลังสูง

2.22 แบตเตอรี่

2.22.1 ชนิด โครงสร้างและขนาดของแบตเตอรี่

2.22.1.1 ซิงค์ (Zinc) ไม่เป็นที่นิยมใช้ในหุ่นยนต์เนื่องจากกำลังงานหมดเร็ว อีกทั้งยังจ่ายพลังงานต่ำ

2.22.1.2 อัลคาไลน์ (Alkaline) ราคาแพงกว่าซิงค์ แต่ให้พลังงานตลอดจนอายุการใช้งานนานกว่า แต่ข้อเสียของแบตเตอรี่แบบนี้ก็คือไม่สามารถประจุไฟฟ้าได้ใหม่

2.22.1.3 นิกเกิล-แคดเมียม (Nickel-Cadmium : Ni-Cad) สามารถประจุใหม่ได้แต่ให้พลังงานน้อยจึงเหมาะที่จะใช้ในหุ่นยนต์ขนาดเล็กใช้งานได้ไม่นาน ในปัจจุบันมีแบตเตอรี่นิกเกิล-แคดเมียมความจุสูงกว่าเดิมประมาณ 2-3 เท่า

2.22.1.4 ตะกั่ว-กรด (Lead – Acid) มีเห็นทั่วไปในรถยนต์ , รถจักรยานยนต์ สามารถประจุได้ใหม่มีทั้งขนาดเล็กและขนาดใหญ่ มีพลังงานสูงแต่มีน้ำหนักมาก ทำให้รถต้องแบกน้ำหนักมากขึ้น นอกจากนี้เราจะเห็นว่ามีการใช้แบตเตอรี่ตะกั่ว-กรดกับระบบจ่ายไฟไม่ขาดตอน (UPS) สำหรับคอมพิวเตอร์หรือระบบโทรศัพท์ สำหรับรถที่ใช้กันอยู่โดยมากจะใช้แบตเตอรี่แบบนี้

2.22.1.5 เจลเซล (Gel – Sel) ประจุใหม่ คล้ายตะกั่ว-กรดแต่จ่ายกระแสสูงกว่า

2.22.2 ข้อกำหนดของแบตเตอรี่

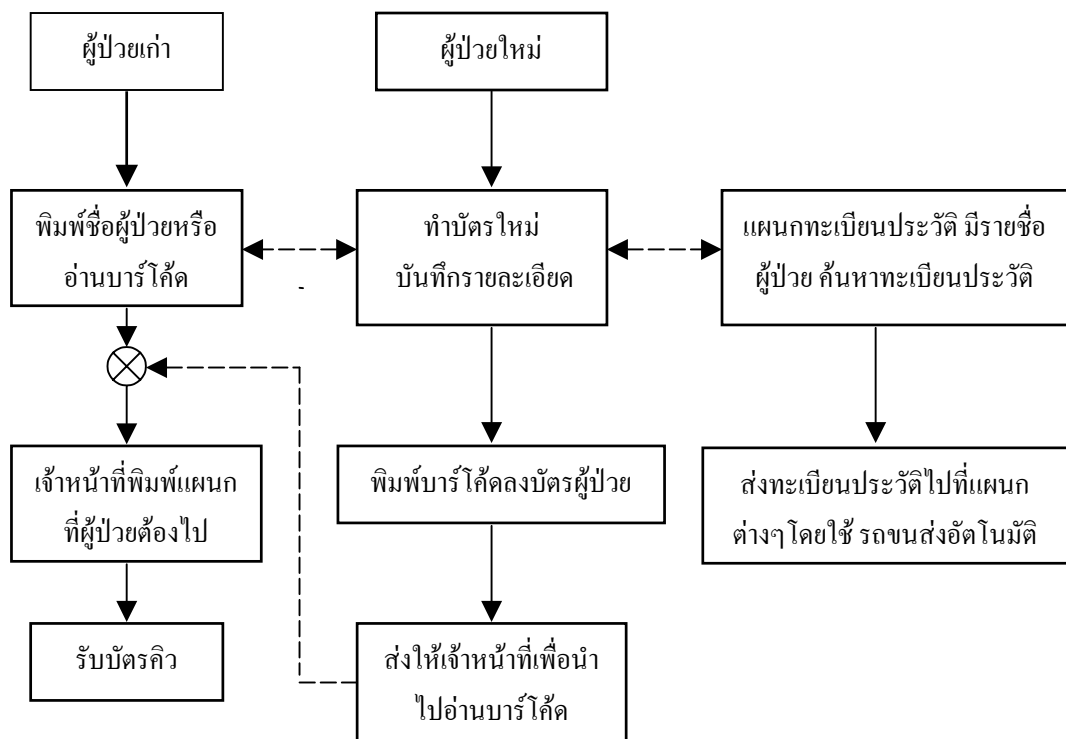
2.22.2.1 แรงดัน สำหรับแบตเตอรี่ที่สามารถประจุใหม่ได้นั้น ขณะประจุเต็มที่จะมีระดับแรงดันสูงกว่าปกติที่กำหนดประมาณ 20 – 30 % เช่น แบตเตอรี่ 12 โวลต์ ชนิดตะกั่ว-กรด เมื่อประจุเต็มที่จะมีค่าประมาณ 13.8 โวลต์ แบตเตอรี่ทั้งหมดจะถือว่าหมดสภาพ ถ้าแรงดันตกต่ำกว่า 4.8 โวลต์ ขณะที่แบตเตอรี่หมดสภาพ มันจะไม่สามารถจ่ายกระแสได้ตามอัตราปกติอีกต่อไป การทดสอบระดับแรงดันแบตเตอรี่ จะต้องทำในขณะที่ใช้งานแบตเตอรี่อยู่ การทดสอบจะคลาดเคลื่อนถ้าขณะทดสอบขณะไม่มีโหลด (Load)

2.22.2.2 กระแส เวลา (Ampere – Hour) แสดงถึงปริมาณความจุของแบตเตอรี่

บทที่ 3 ขั้นตอนการดำเนินงาน

3.1 ขั้นตอนการทำงานร่วมกันของระบบฐานข้อมูลแผนกเวชระเบียนกับรถขนส่งอัตโนมัติ

การทำงานร่วมกันของระบบฐานข้อมูลแผนกเวชระเบียนกับรถขนส่งอัตโนมัติ แสดงได้จาก รูปที่ 3.1 ในกรณีผู้ป่วยเก่าเจ้าหน้าที่จะพิมพ์ชื่อหรืออ่านบาร์โค้ดที่บัตรประจำตัวผู้ป่วยเพื่อค้นหาประวัติ ส่วนผู้ป่วยใหม่เจ้าหน้าที่จะบันทึกประวัติแล้วทำการพิมพ์บาร์โค้ดลงบัตรประจำตัวผู้ป่วย ขั้นตอนต่อไปเจ้าหน้าที่จะพิมพ์แผนกที่คนไข้จะต้องไปและจัดการพิมพ์บัตรคิว ซึ่งในส่วนนี้ทั้งหมดจะทำงานผ่านระบบฐานข้อมูล จากนั้นเจ้าหน้าที่แผนกทะเบียนประวัติจะส่งทะเบียนประวัติผู้ป่วยไปยังแผนกต่างๆที่ผู้ป่วยจะเข้าทำการรักษาโดยรถขนส่งอัตโนมัติ



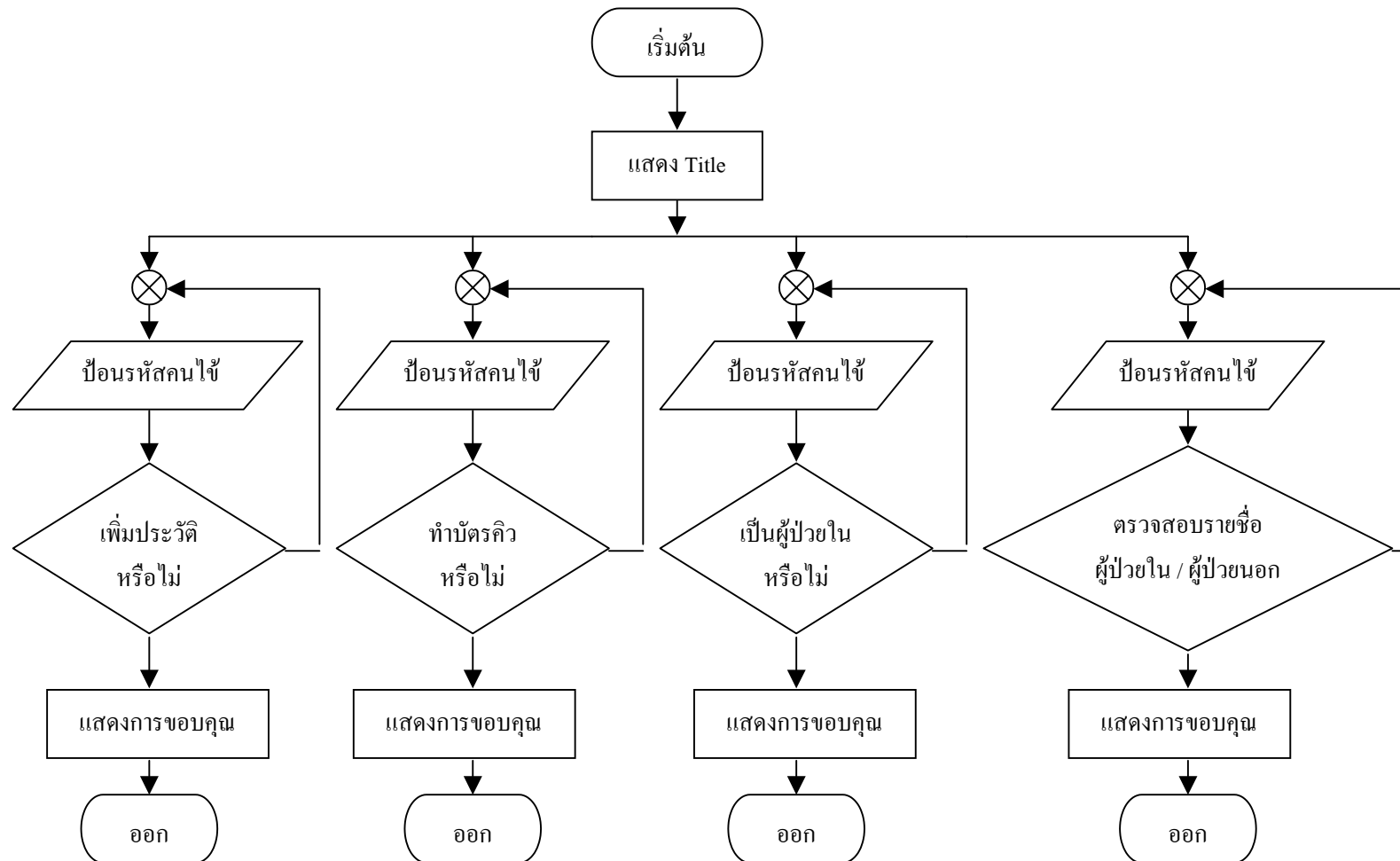
รูปที่ 3.1 ब्लॉकไดอะแกรมการทำงานร่วมกันระหว่างระบบฐานข้อมูลและรถขนส่งอัตโนมัติ

3.2 ขั้นตอนการทำงานของระบบฐานข้อมูล

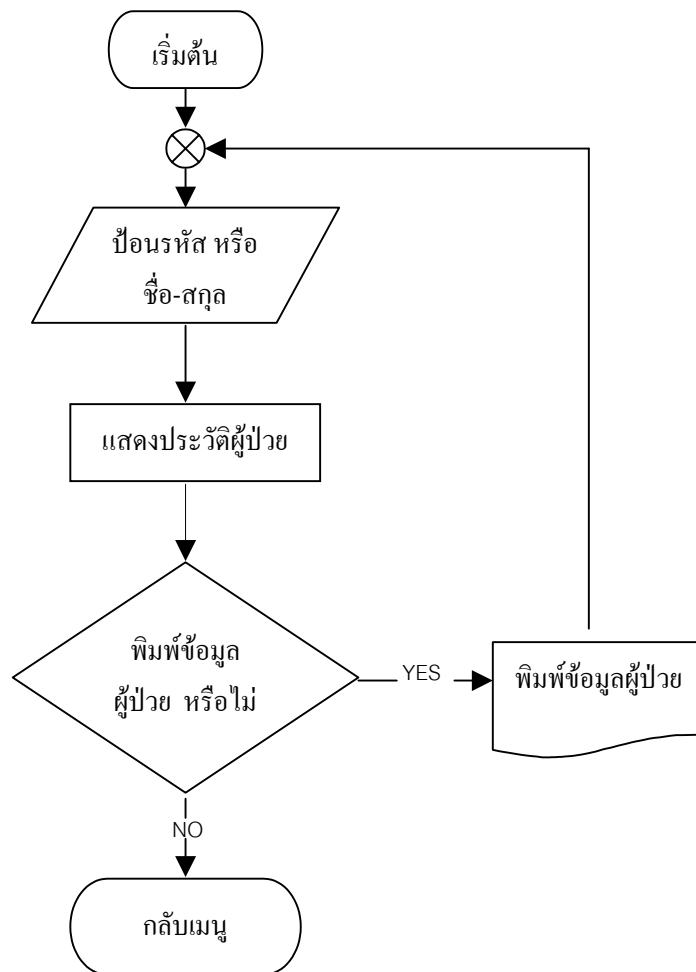
การนำระบบคอมพิวเตอร์เข้ามาประยุกต์ใช้ร่วมกับระบบบาร์โค้ด จะมีส่วนช่วยในการจัดระบบการบันทึกประวัติการค้นหารายชื่อคนไข้ การจัดลำดับบัตรคิว ทำให้ใช้เวลาในการทำงานที่น้อยลงและมีความถูกต้องในการทำงานเพิ่มขึ้นจากปัญหาข้างต้นซึ่งได้กล่าวไปแล้วแนวทางแก้ไขปัญหาทำได้ดังนี้

3.2.1 นำระบบจัดการข้อมูล เข้ามาใช้ทำให้ข้อมูลที่เก็บมีความ เป็นระเบียบค้นหาได้ง่าย และสามารถสำรองข้อมูลได้ ด้วยแผ่นดิสก์เพียงไม่กี่แผ่น ใช้ระบบฐานข้อมูลเชื่อมโยงการทำงานด้วยระบบบาร์โค้ด ซึ่งช่วยลดเวลาในการทำงานและข้อผิดพลาดก็น้อยลง

3.2.2 ขั้นตอนการดำเนินงานในการจัดสร้างโปรแกรมจัดการฐานข้อมูล มีขั้นตอนการทำโครงการโปรแกรมจัดการฐานข้อมูลด้วยระบบบาร์โค้ดซึ่งมีรายละเอียดของการดำเนินงานดังโพลีชาร์ต เมื่อเข้าสู่โปรแกรมการทำงานของฐานข้อมูลแล้ว จะมีเมนูคำสั่งเข้ามาให้เลือกทำการไปยังส่วนต่างๆ ทั้งการดูประวัติของผู้ป่วย การจัดทำบัตรคิว การจัดผู้ป่วยเข้าเป็นผู้ป่วยใน การตรวจสอบรายชื่อผู้ป่วยใน/ผู้ป่วยนอก ซึ่งสามารถเข้าไปยังส่วนต่างๆ ได้โดยทำการกดปุ่ม (Click) ที่หน้าจอ และเมื่อจบการทำงานของโปรแกรมแล้วจะมีข้อความแสดงคำขอบคุณ จะดูได้ดังรูปที่ 3.2

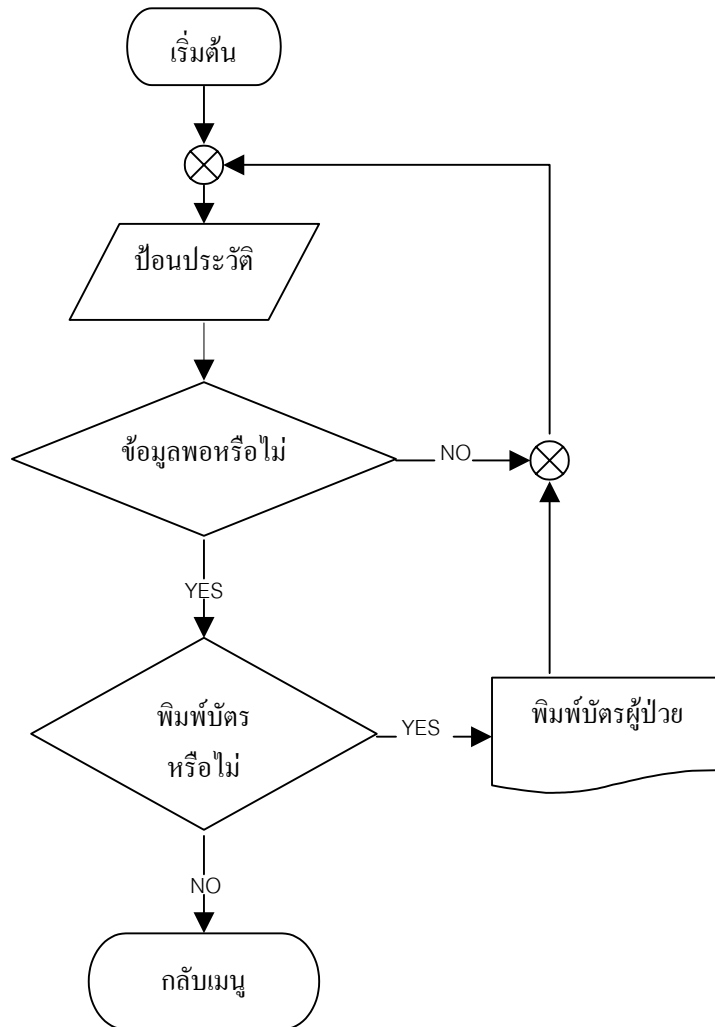


รูปที่ 3.2 บล็อกไดอะแกรมการเริ่มทำงานเมื่อเปิดโปรแกรมฐานข้อมูล



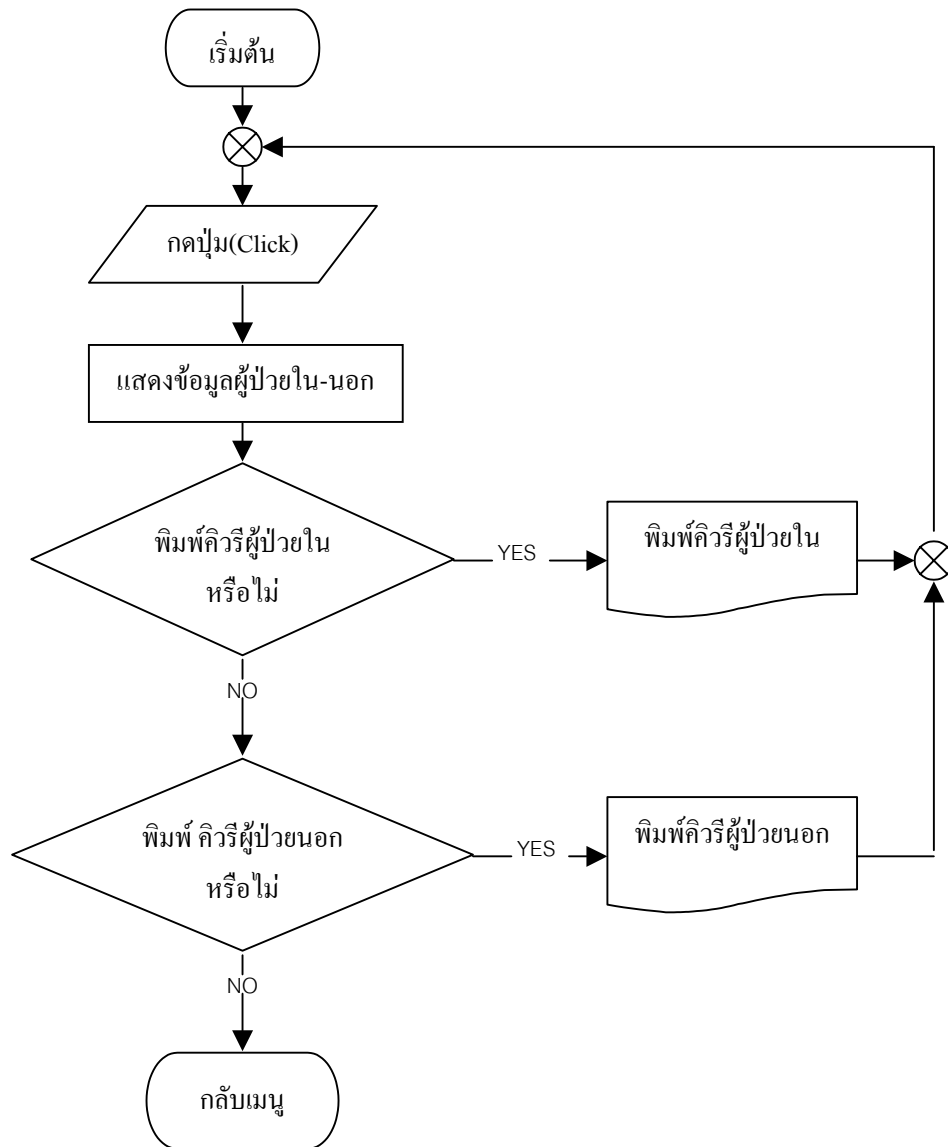
รูปที่ 3.3 โฟลว์ชาร์ตการดูประวัติผู้ป่วย

ขั้นตอนเริ่มด้วยการป้อนรหัส หรือชื่อ-สกุล เข้าไปโปรแกรมจะทำการค้นหาข้อมูลของผู้ป่วย จากนั้นก็จะแสดงรายละเอียดของประวัติผู้ป่วยที่มีออกมาและสามารถทำการพิมพ์ออกมาได้ในกรณีที่ต้องการ



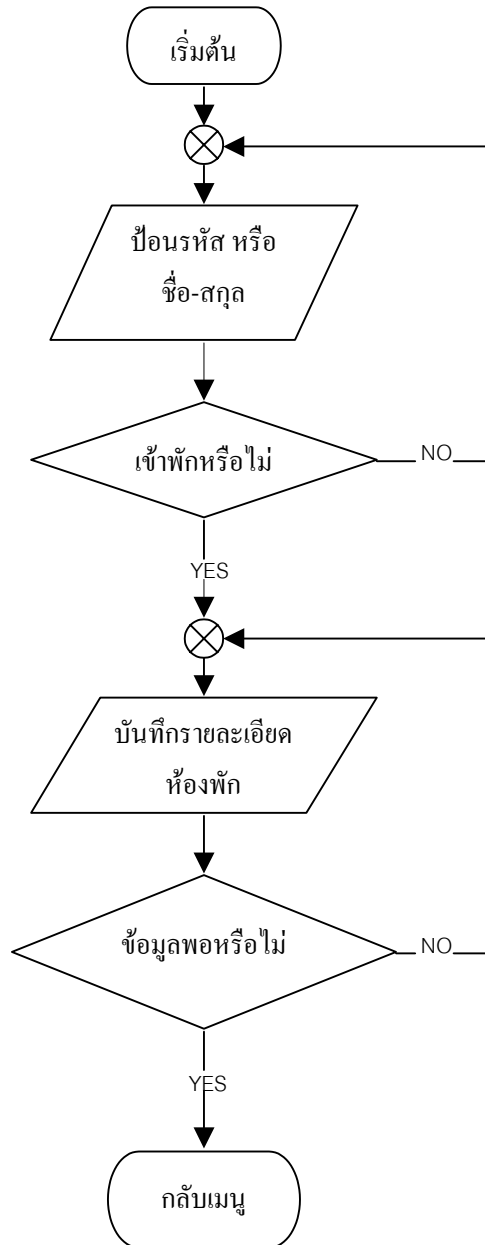
รูปที่ 3.4 โฟลว์ชาร์ตการกรอกประวัติคนไข้ใหม่

ขั้นตอนเริ่มจากการกรอกประวัติคนไข้ใหม่ เข้าไป และยังสามารถแก้ไขประวัติคนไข้ตรงส่วนนี้ได้ ด้วยจากนั้นก็ทำการบันทึกเข้าไปโดยคำสั่งบันทึก (Save) และสามารถที่จะพิมพ์บัตรผู้ป่วยใหม่ที่ยังไม่มิดได้ และนำมาติดกับรหัสบาร์โค้ดที่มีค่าเดียวกับรหัสผู้ป่วย



รูปที่ 3.5 โฟลว์ชาร์ตการขอดูรายละเอียดผู้ป่วยใน / นอก

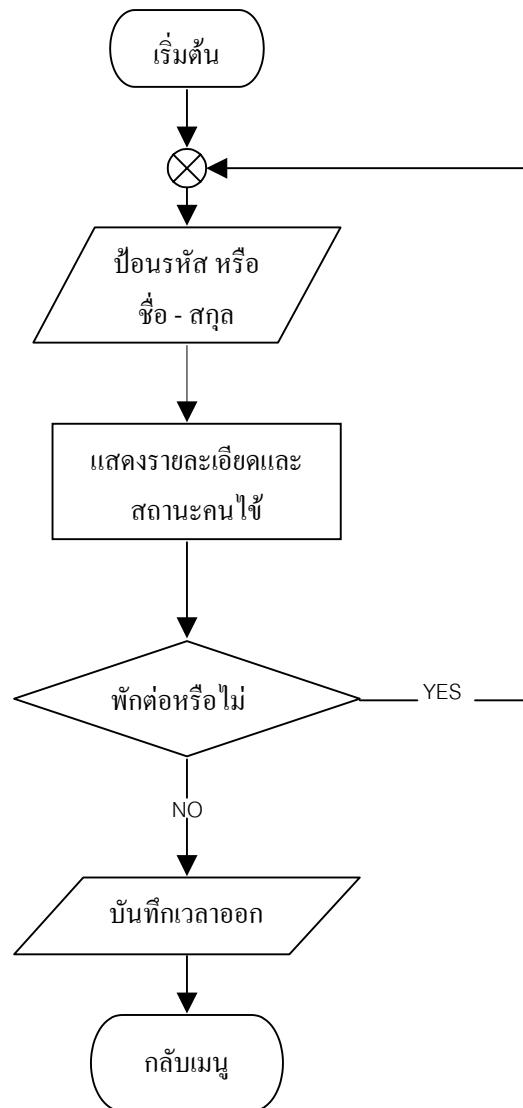
เมื่อทำการกดปุ่ม (Click) ที่หน้าเมนู เพื่อเข้ามาขอดูรายละเอียดของผู้ป่วยใน / นอกทั้งหมด โปรแกรมก็ทำการแสดงออกมาโดยแบ่งเป็นส่วนของผู้ป่วยใน และส่วนของผู้ป่วยนอก จากนั้นจะมีปุ่มสั่งพิมพ์รายชื่อทั้งหมดของแต่ละส่วน ถ้าต้องการก็สามารถพิมพ์ออกมาได้



รูปที่ 3.6 โฟลว์ชาร์ตการจัดคนไข้เป็นผู้ป่วยใน

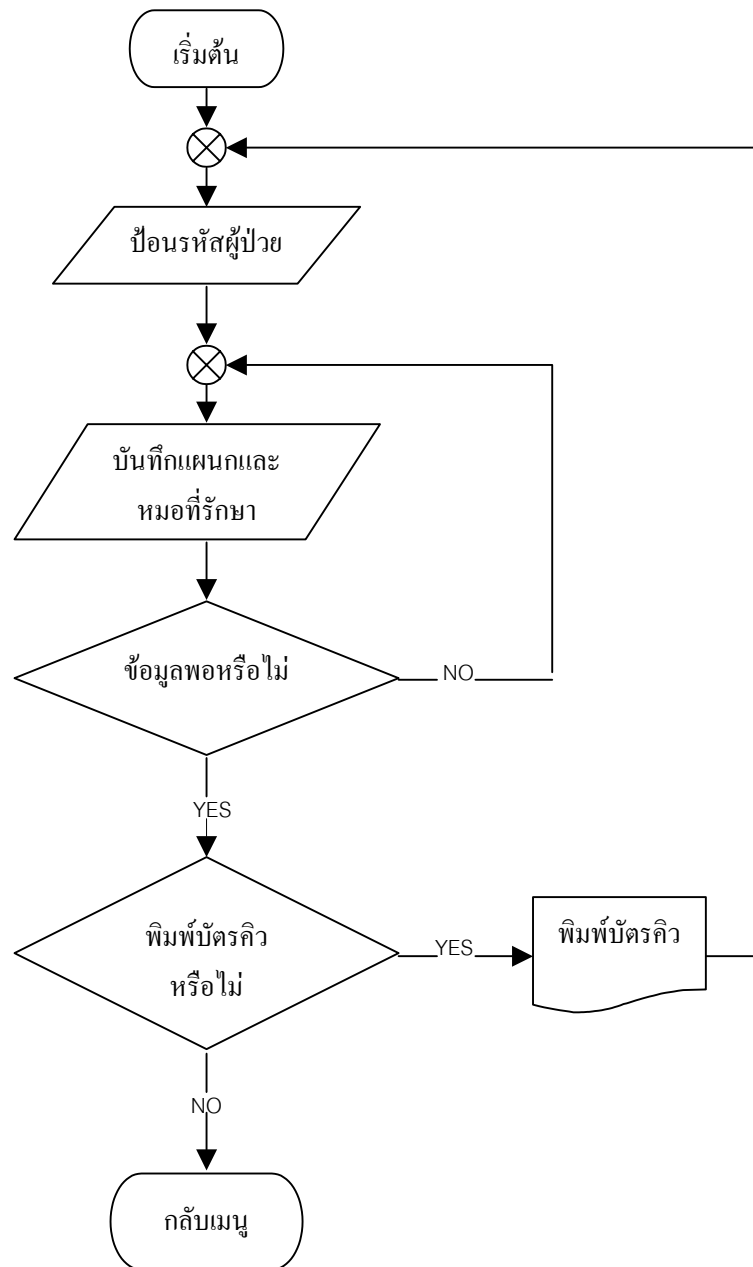
ขั้นตอนเมื่อทำการป้อนรหัส หรือ ชื่อ-สกุล แล้วระบบทำการค้นหารหัส หรือ ชื่อ-สกุล เมื่อพบและแสดงรายละเอียด จะมีปุ่มรายละเอียดเพิ่มเติมสำหรับผู้ป่วยที่จะจัดเข้าเป็นผู้ป่วยใน เมื่อเข้าไปในส่วนนี้ก็จะมีการให้บันทึกรายละเอียดของ

ห้องพัก วันเวลาที่เข้าพัก และทำการบันทึกลงไปได้ จากนั้นข้อมูลส่วนนี้ก็จะไปแสดงในส่วนจของรายละเอียดผู้ป่วยในทั้งหมด



รูปที่ 3.7 โฟลว์ชาร์ตการเช็คเอาท์ (Check Out) ของผู้ป่วยใน

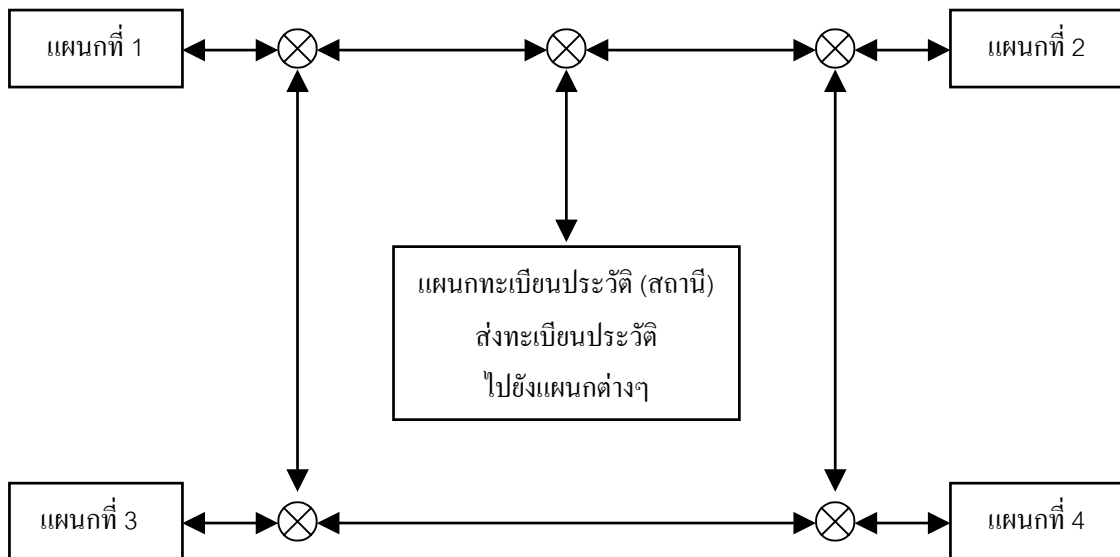
เมื่อถึงเวลาที่ ผู้ป่วยจะทำการเช็คเอาท์ (Check Out) ให้ทำการป้อนรหัสหรือ ชื่อ -สกุล จากนั้นเข้าไปตรงส่วนการเช็คเอาท์เพื่อทำการเลือกให้ผู้ป่วยนี้กลายเป็นผู้ป่วยนอก และระบบจะทำการบันทึกเวลาออกล่าสุดที่ออกจากโรงพยาบาล ในส่วนที่แสดงรายละเอียดของผู้ป่วยใน / นอก ผู้ป่วยคนนั้นก็กลายเป็นผู้ป่วยนอก และจะแสดงวัน เวลาครั้งล่าสุดที่ เคยเข้าพักในโรงพยาบาล



รูปที่ 3.8 โฟลว์ชาร์ตการจัดลำดับบัตรคิว

ในส่วนนี้กรณีการจัดลำดับบัตรคิวให้ผู้ป่วย เพื่อส่งไปยังแต่ละแผนกเมื่อทำการป้อนรหัสผู้ป่วย หรือ ชื่อ-สกุล จากนั้นเจ้าหน้าที่จะทำการบันทึกแผนก และแพทย์ที่จะทำการรักษา และระบุห้องที่จะให้แพทย์ตรวจคนไข้เมื่อทำการบันทึกเสร็จแล้วจึงทำการสั่งพิมพ์บัตรคิว ซึ่งในบัตรคิวจะแสดงลำดับที่จะเข้าพบแพทย์ของผู้ป่วย

3.3 ขั้นตอนการทำงานของรถขนส่งอัตโนมัติ



สมมติให้ แผนกที่ 1 เป็น แผนกกุมารเวช

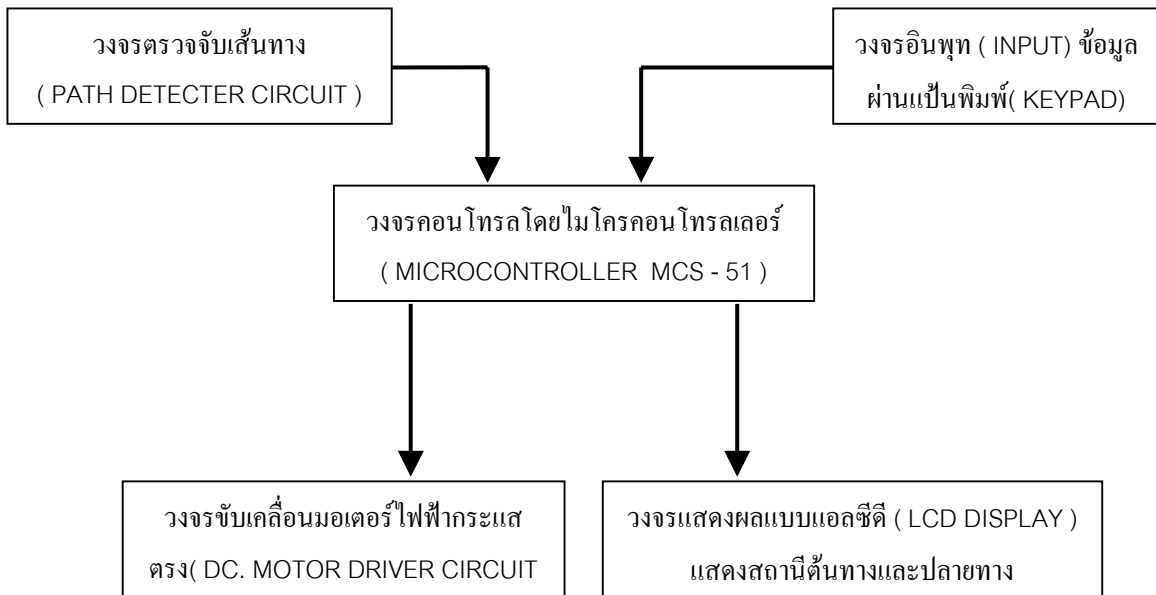
แผนกที่ 2 เป็น แผนกทันตกรรม

แผนกที่ 3 เป็น แผนกจักษุเวช

แผนกที่ 4 เป็น แผนกศัลยกรรม

รูปที่ 3.9 บล็อกไดอะแกรมการทำงานของรถขนส่ง

เมื่อเจ้าหน้าที่ทราบว่าจะส่งทะเบียนประวัติผู้ป่วยไปยังแผนกใดแล้ว ก็จะทำการป้อนคำสั่งที่เป็นตัวเลขของแผนกต้นทางและแผนกปลายทางผ่านทางแป้นพิมพ์ (Keypad) ที่อยู่บนตัวรถขนส่งอัตโนมัติ รถขนส่งอัตโนมัติจะวิ่งไปจนถึงสถานีปลายทางแล้วจะหยุด จนกว่าจะมีเจ้าหน้าที่มาป้อนคำสั่งให้ทำงานอีกครั้ง



รูปที่ 3.10 บล็อกไดอะแกรมการเชื่อมโยงวงจรของรถขนส่งอัตโนมัติ

เมื่อป้อนคำสั่งที่เป็นตัวเลขของแผนกต้นทางและปลายทางผ่านทางแป้นพิมพ์ (Keypad) แล้วไมโครคอนโทรลเลอร์จะเริ่มรับค่าจากวงจรตรวจจับเส้นทางมาประมวลผล แล้วส่งค่าเอาต์พุท (Output) ที่เป็น ลอจิก (Logic) ไปยังวงจรขับเคลื่อนมอเตอร์ เพื่อบังคับให้รถขนส่งอัตโนมัติเดินไปตามทิศทางที่เรา กำหนดไว้ได้อย่างถูกต้อง

3.3.1 การทำงานของวงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง (DC.Motor Driver Circuit)

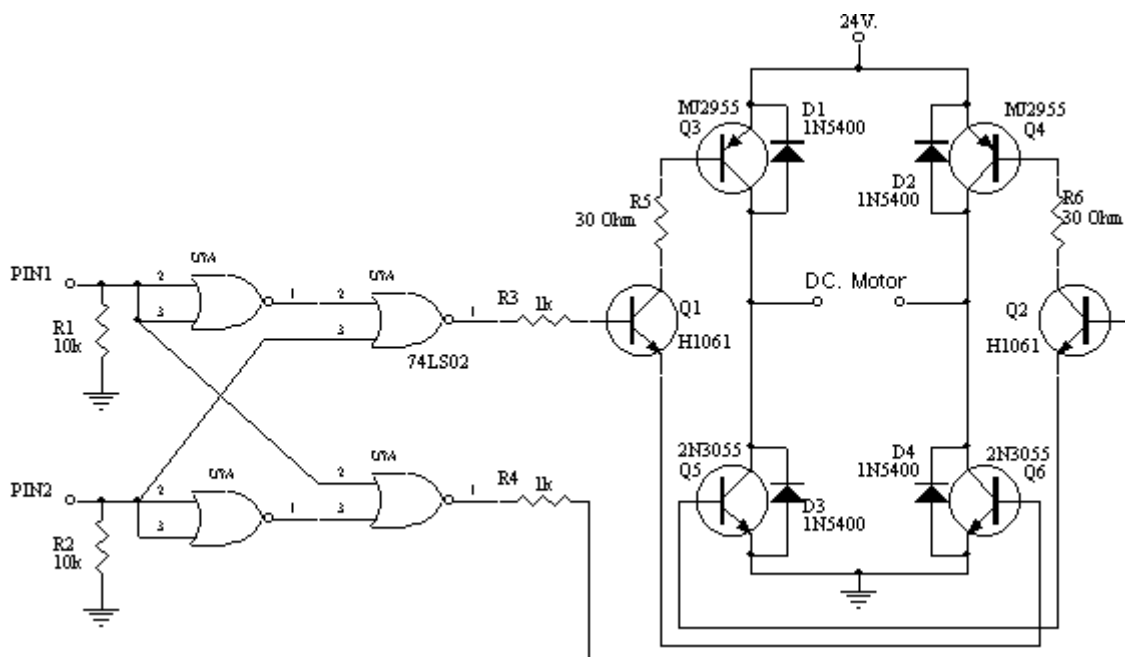
วงจรมีหน้าที่ในการขับเคลื่อนให้มีสถานะการหมุนของมอเตอร์แบบเดินหน้า (Forward) หรือถอยหลัง (Reverse) โดยใช้วงจรควบคุมมอเตอร์กระแสตรงแบบ 4 ควอดแรนต์ (4-Quadrant) ซึ่งจะใช้ เพาเวอร์ทรานซิสเตอร์ (Power Transistor) เป็นตัวเปลี่ยนทิศทางการไหลของกระแสที่ผ่านมอเตอร์ในการทำงานของวงจรมีการรับสัญญาณควบคุม (Input Signal) อยู่ 2 สัญญาณคือสัญญาณจาก PIN1 และ PIN2 โดยจะนำสัญญาณควบคุมทั้ง 2 สัญญาณต่อเข้ากับวงจรรนอร์เกต (Nor-Gate) เพื่อป้องกันความผิดพลาดเมื่อสัญญาณควบคุมซึ่งอาจทำให้วงจรถับเคลื่อนเสียหาย เนื่องไปการทำงานของลอจิกจากไมโครคอนโทรลเลอร์ มี 4 กรณี ดังนี้

3.3.1.1 PIN1 มีลอจิกเป็น 1 และ PIN2 มีลอจิกเป็น 0 จะทำให้ Q1 ทำหน้าที่ไบอัส (Bias) ที่ขาเบสของ Q3 และ Q6 กระแสจะไหลผ่าน Q3 ไปยัง มอเตอร์และผ่าน Q6 ไปยังกราวด์ (Ground) เป็นผลให้มอเตอร์หมุนในแบบเดินหน้า (Forward)

3.3.1.2 PIN1 มีลอจิกเป็น 0 และ PIN2 มีลอจิกเป็น 1 จะทำให้ Q2 ทำหน้าที่ไบอัส (Bias) ที่ขาเบสของ Q4 และ Q5 กระแสจะไหลผ่าน Q4 ไปยัง มอเตอร์และผ่าน Q5 ไปยังกราวด์ (Ground) เป็นผลให้มอเตอร์หมุนในแบบถอยหลัง (Reverse)

3.3.1.3 PIN1 และ PIN2 มีลอจิกเป็น 1 มอเตอร์จะไม่ทำงาน

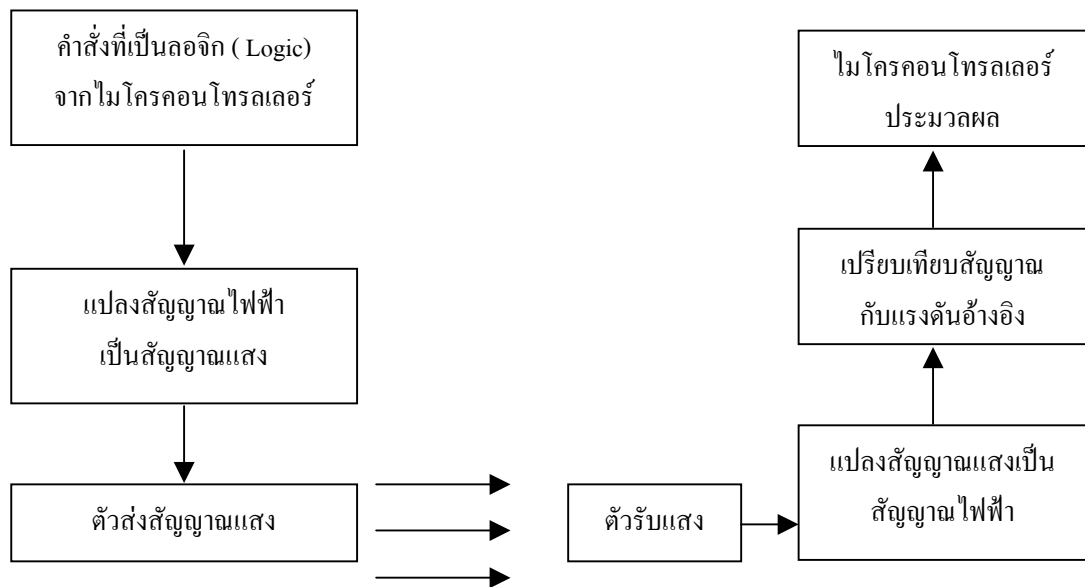
3.3.1.4 PIN1 และ PIN2 มีลอจิกเป็น 0 มอเตอร์จะไม่ทำงาน



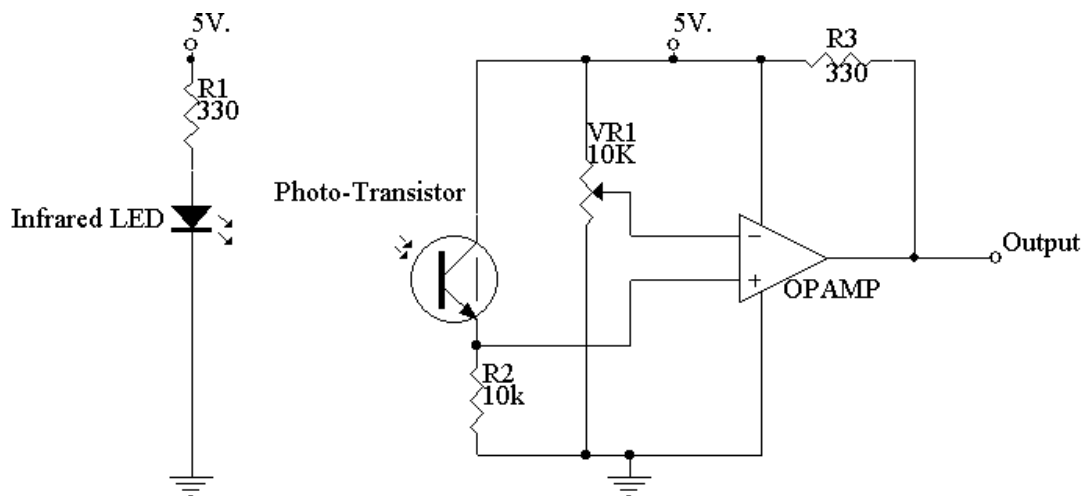
รูปที่ 3.11 วงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง

3.3.2 วงจรตรวจจับเส้นทาง (Path Detector Circuit)

วงจรตรวจจับเส้นทางแบ่งออกเป็น 2 ภาคด้วยกัน คือ ภาคส่งและภาครับสัญญาณแสงอินฟราเรด ซึ่งมีการทำงานร่วมกันแสดงได้จากรูปที่ 3.12 และ 3.13



รูปที่ 3.12 บล็อกไดอะแกรมการทำงานร่วมกันของภาคส่งและภาครับแสงอินฟราเรด
ของวงจรตรวจจับเส้นทาง (Path Detector Circuit)



รูปที่ 3.13 วงจรภาคส่งและภาครับแสงอินฟราเรดของวงจรตรวจจับเส้นทาง (Path Detector Circuit)

3.3.2.1 ภาคส่งสัญญาณแสงอินฟราเรด

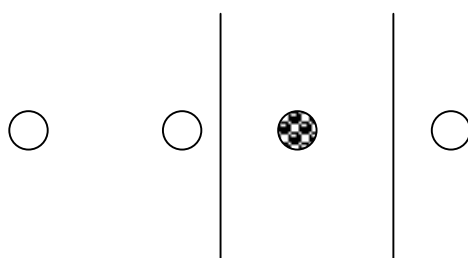
การทำงานของภาคส่งสัญญาณแสงอินฟราเรดเราจะจ่ายแรงดันให้กับไดโอดเปล่งแสงอินฟราเรด (High Power Near-Infrared Emitting Diode) เพื่อให้เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานแสง

3.3.2.2 ภาครับสัญญาณแสงอินฟราเรด

วงจรับสัญญาณแสงอินฟราเรดอาศัยหลักการตอบสนองต่อแสงของ โฟโตทรานซิสเตอร์ (Photo-Transistor) เปลี่ยนพลังงานแสงให้เป็นแรงดันไฟฟ้า และส่งต่อไปที่ออปแอมป์ (Opamp) เพื่อเปรียบเทียบกับแรงดันอ้างอิงที่เรากำหนดไว้ เอาท์พุท (Output) ที่ได้จะเป็น 0 V หรือ 5 V ซึ่งก็คือ ลอจิก 0 หรือ ลอจิก 1 แล้วส่งไปที่ไมโครคอนโทรลเลอร์เพื่อประมวลผลต่อไป

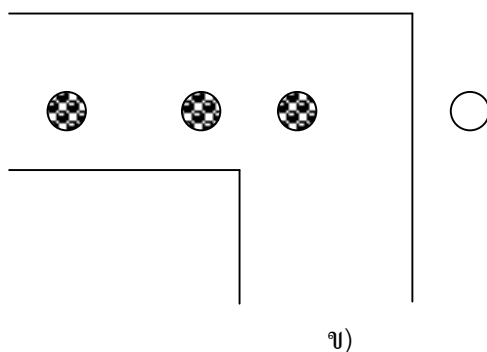
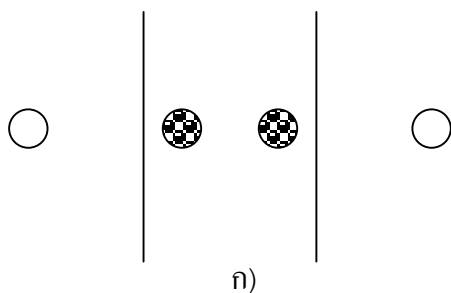
3.3.3 ตำแหน่งของเซ็นเซอร์ (Sensor) เมื่อวงจรตรวจจับเส้นทางทำงาน

3.3.3.1 เมื่อรถขนส่งอัตโนมัติวิ่งทางตรงตำแหน่งของเซ็นเซอร์จะเป็นดังรูปที่ 3.14



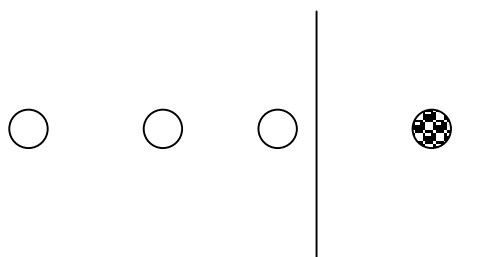
รูปที่ 3.14 ตำแหน่งของเซ็นเซอร์เมื่อรถขนส่งอัตโนมัติวิ่งทางตรง

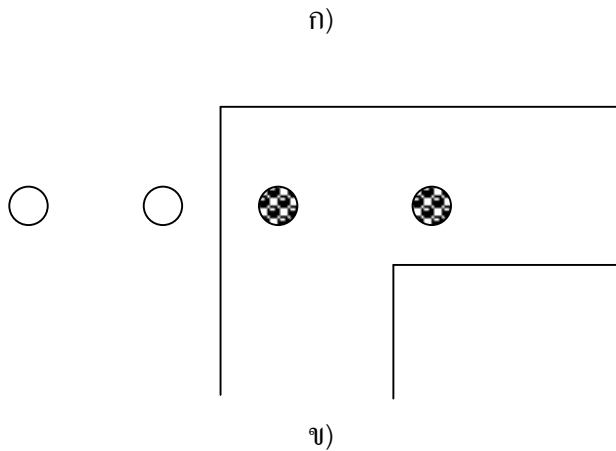
3.3.3.2 เมื่อเซ็นเซอร์ของรถขนส่งอัตโนมัติเป็นดังรูปที่ 3.15 ก) และ ข) วงจรตรวจจับเส้นทางจะส่งสัญญาณที่เป็นลอจิกไปยังไมโครคอนโทรลเลอร์เพื่อประมวลผล แล้วส่งสัญญาณควบคุมไปที่วงจรขับเคลื่อนมอเตอร์สั่งให้รถขนส่งอัตโนมัติเลี้ยวซ้าย เพื่อไม่ให้รถขนส่งอัตโนมัติวิ่งออกนอกเส้นทาง



รูปที่ 3.15 ตำแหน่งของเซ็นเซอร์ที่รถขนส่งอัตโนมัติต้องเลี้ยวซ้าย

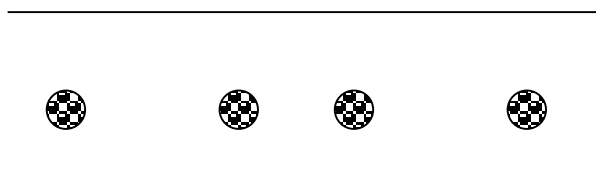
3.3.3.3 เมื่อเซ็นเซอร์ของรถขนส่งอัตโนมัติเป็นดังรูปที่ 3.16 ก) และ ข) วงจรตรวจจับเส้นทางจะส่งสัญญาณที่เป็นลอจิกไปยังไมโครคอนโทรลเลอร์เพื่อประมวลผล แล้วส่งสัญญาณควบคุมไปที่วงจรขับเคลื่อนมอเตอร์สั่งให้รถขนส่งอัตโนมัติเลี้ยวขวา เพื่อไม่ให้รถขนส่งอัตโนมัติวิ่งออกนอกเส้นทาง





รูปที่ 3.16 ตำแหน่งของเซ็นเซอร์ที่รถยนต์อัตโนมัติต้องเลี้ยวขวา

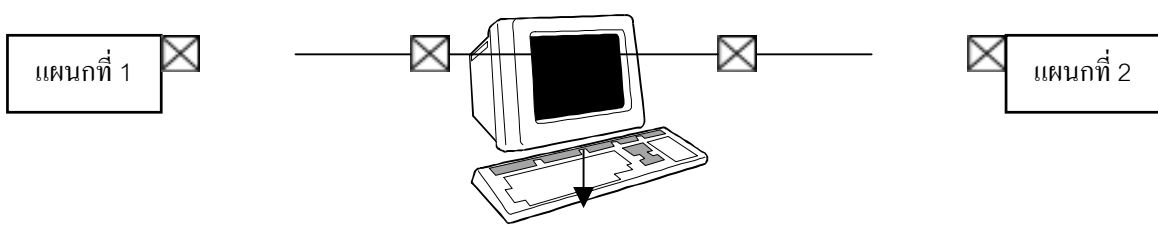
3.3.3.4 เมื่อเซ็นเซอร์ของรถยนต์อัตโนมัติเป็นดังรูปที่ 3.17 วงจรตรวจจับเส้นทางจะส่งสัญญาณที่เป็นลอจิกไปยังไมโครคอนโทรลเลอร์เพื่อประมวลผล แล้วส่งสัญญาณควบคุมไปที่วงจรขับเคลื่อนมอเตอร์สั่งให้รถยนต์อัตโนมัติเลี้ยวซ้ายหรือเลี้ยวขวาตามเส้นทางไปสู่สถานีปลายทาง หรือสั่งให้รถยนต์อัตโนมัติหยุดเมื่อถึงสถานีปลายทาง



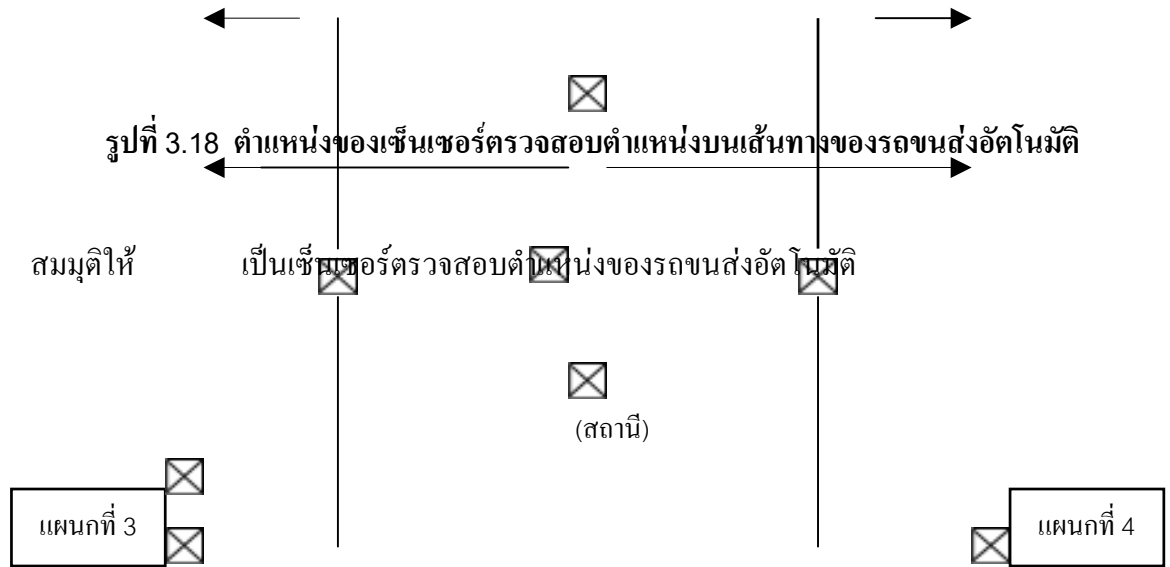
รูปที่ 3.17 ตำแหน่งของเซ็นเซอร์ที่รถยนต์อัตโนมัติต้องเลี้ยวซ้ายหรือเลี้ยวขวาหรือหยุด

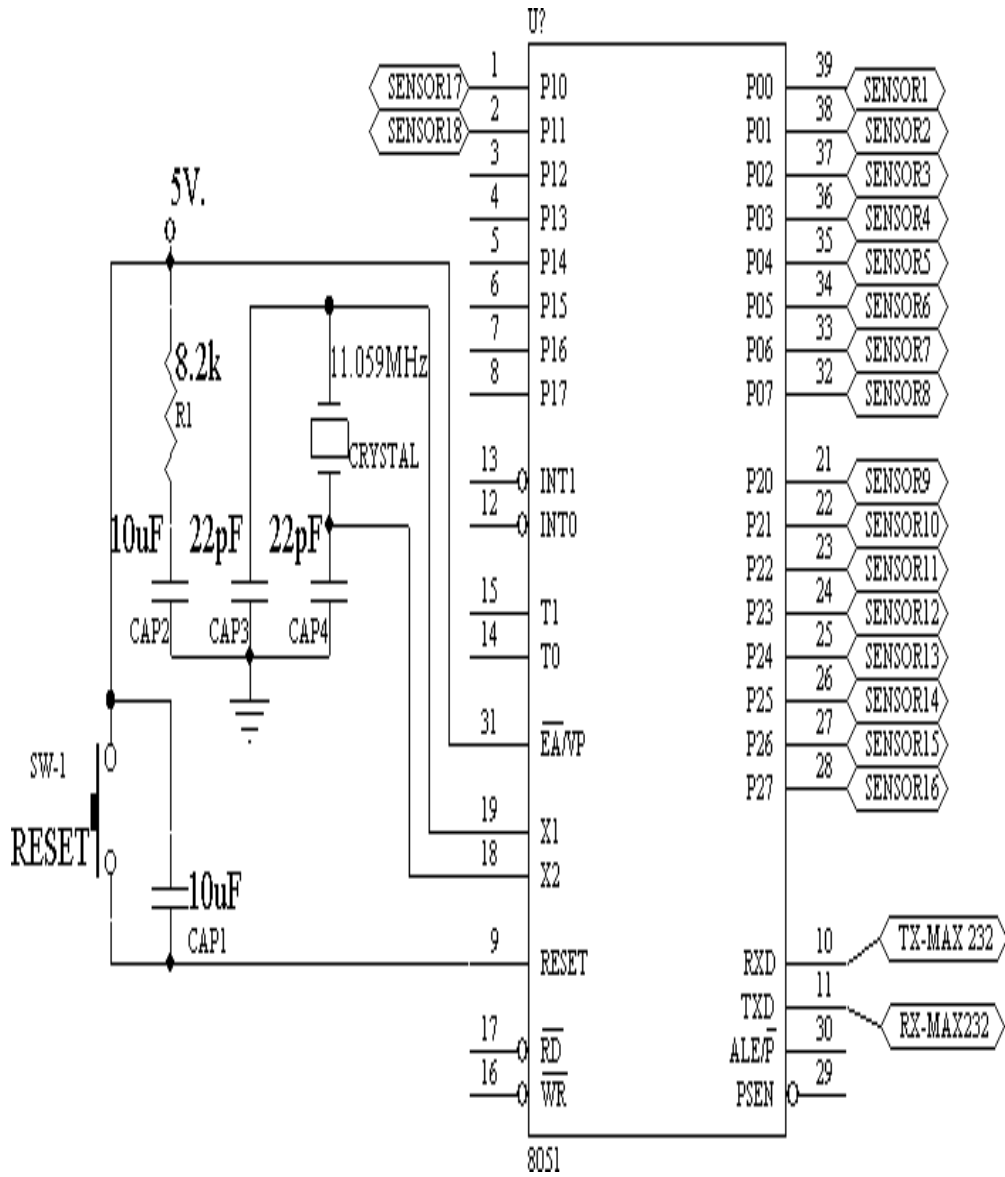
3.4 ระบบตรวจสอบตำแหน่งของรถยนต์อัตโนมัติ

เมื่อรถยนต์อัตโนมัติเคลื่อนที่ออกจากแผนกต้นทางเพื่อไปยังแผนกปลายทาง ในช่วงระหว่างที่เดินทางนั้นอาจมีการผิดพลาดขึ้นได้ เช่น แบตเตอรี่หมดทำให้รถหยุดวิ่ง หรือ เส้นทางสกปรกทำให้เซ็นเซอร์ตรวจจับเส้นทางทำงานผิดพลาดจนรถยนต์อัตโนมัติหลุดออกนอกเส้นทาง จึงจำเป็นต้องมีระบบตรวจสอบตำแหน่งของรถยนต์อัตโนมัติ โดยจะมีการติดตั้งวงจรตรวจสอบตำแหน่งของรถยนต์อัตโนมัติไว้เป็นระยะตลอดเส้นทางและที่แต่ละแผนก และมีการแสดงตำแหน่งของรถยนต์อัตโนมัติผ่านทางหน้าจอคอมพิวเตอร์ ตำแหน่งของเซ็นเซอร์ตรวจสอบตำแหน่งแสดงได้ดังรูปที่ 3.18 และวงจรที่

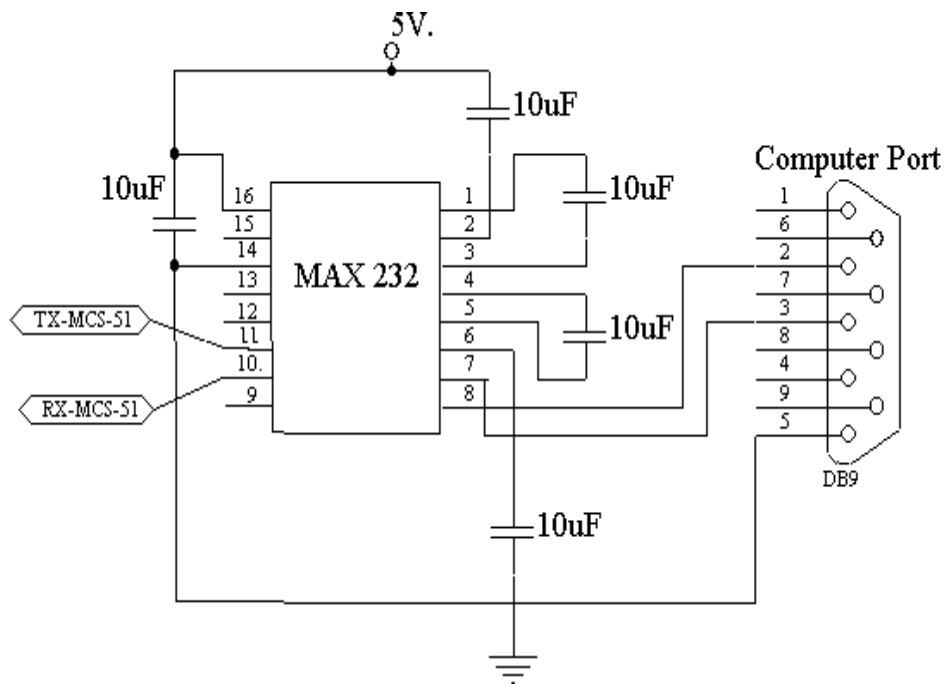


ใช้ติดต่อระหว่างเซ็นเซอร์ตรวจสอบตำแหน่งกับคอมพิวเตอร์แสดงได้ดังรูปที่ 3.19 ก) และ ข)





n)



ข)

รูปที่ 3.19 แสดงวงจรรับค่าจากเซ็นเซอร์ตรวจสอบตำแหน่ง
ของรถขนส่งอัตโนมัติเพื่อเชื่อมต่อกับคอมพิวเตอร์

จากรูปที่ 3.19 ก) เมื่อเซ็นเซอร์ตรวจจับเส้นทางที่ติดตั้งไว้ตามจุดต่างๆ ของเส้นทางตรวจจับได้ว่ารถขนส่งอัตโนมัติวิ่งผ่านไป เซ็นเซอร์จะส่งสัญญาณไปเป็นอินพุต (Input) ให้ไมโครคอนโทรลเลอร์ MCS-51 ประมวลผล และส่งสัญญาณไปยังไอซี แมกซ์ 232 (MAX 232) ดังรูปที่ 3.19 ข) เพื่อเป็นการส่งต่อข้อมูลจากภายนอก เข้าสู่คอมพิวเตอร์ (Interface) และ แสดงผลทางหน้าจอด้วยโปรแกรมวิซวลเบสิก (Visual Basic)

บทที่ 4

ผลการทดลอง

4.1 ผลการทดลองการจัดการฐานข้อมูลด้วยบาร์โค้ด (Barcode)

จากการจัดทำฐานข้อมูล (Database) และแอปพลิเคชัน (Application) ในไมโครซอฟท์ แอ็กเซส (Microsoft Access) แล้วจะได้ฐานข้อมูลและหน้าจอที่ใช้ในการกรอกข้อมูลต่างๆดังต่อไปนี้

4.1.1 ฟอรัมเมนูหลัก



รูปที่ 4.1 หน้าจอเมนูหลัก

การทำงานของฟอร์มเมนูหลักนี้ จะเป็นหน้าจอแรกเมื่อ โปรแกรมเริ่มทำงาน ในฟอร์มการทำงานนี้จะมีปุ่มต่างๆเพื่อใช้งานดังนี้

4.1.1.1 ปุ่มเพิ่มประวัติผู้ป่วยใหม่ เมื่อกดปุ่มนี้โปรแกรมจะแสดงหน้าจอตั้งในรูปที่ 4.2 เพื่อกรอกข้อมูลคนไข้ใหม่ที่เข้ามาทำการรักษาเข้าไปในฐานข้อมูล

4.1.1.2 ปุ่มแก้ไขประวัติผู้ป่วยใช้เพื่อแก้ไขประวัติของคนไข้เก่า

4.1.1.3 ปุ่มจัดผู้ป่วยในเข้า-ออก ใช้เมื่อต้องการจัดให้คนไข้ที่เข้ามาพักในโรงพยาบาล รวมทั้งการจัดคนไข้ที่ออกเมื่อจะออกจากโรงพยาบาลด้วย

4.1.1.4 ปุ่มค้นหาประวัติ จะแบ่งเป็นการค้นหาด้วยชื่อ-สกุล(ที่แน่ใจว่าถูกต้อง) กับการค้นหาประวัติด้วยเลขประจำตัวที่ได้จากการสแกนบาร์โค้ดจากบัตรผู้ป่วย

4.1.1.5 ปุ่มค้นหาผู้ป่วยใน จะแบ่งเป็นการค้นหาด้วยชื่อ-สกุล(ที่แน่ใจว่าถูกต้อง) กับการค้นหาผู้ป่วยในด้วยเลขประจำตัวที่ได้จากการอ่านบาร์โค้ดจากบัตรผู้ป่วย


4.1.1.6 ปุ่มค้นหาผู้ป่วยใน(พิเศษ) จะใช้ในกรณีที่ชื่อ-สกุลที่ค้นหานั้นไม่แน่ใจว่าถูกต้อง โปรแกรมจะทำการหาค่าที่ใกล้เคียงกับค่าที่กรอกเข้าไปแล้วนำมาแสดงผล

4.1.1.7 ปุ่มตั้งค่าเครื่องพิมพ์จะใช้ในการตั้งค่าเครื่องพิมพ์ที่นำมาใช้ในการพิมพ์ข้อมูลที่ต้องการ

4.1.1.8 ปุ่มออกจากโปรแกรม ใช้เมื่อทำงานเสร็จสิ้นแล้วและต้องการออกจากโปรแกรม เครื่องจะถามความแน่ใจและถ้าออกก็จะทำการบันทึกข้อมูลล่าสุดทุกค่าที่มีการเปลี่ยนแปลงเข้าไปในฐานข้อมูล

4.1.2 ฟอร์มการเพิ่มประวัติผู้ป่วยใหม่

หน้าจอนี้จะปรากฏเมื่อกดปุ่มเพิ่มประวัติผู้ป่วยใหม่ ดังรูปที่ 4.2 เมื่อหน้าจอนี้ปรากฏก็สามารถให้ทำการกดปุ่มเพิ่มประวัติผู้ป่วยใหม่ก็จะขึ้นเป็นค่าใหม่โดยสามารถกลับไปดูได้ว่าหมายเลขสุดท้ายของฐานข้อมูลนั้นถึงไหนแล้วจากนั้นก็สามารกรอกประวัติผู้ป่วยที่ต้องการลงไปได้ทันที เมื่อเสร็จแล้วก็ทำการกดปุ่มบันทึกค่าก็จะเก็บเข้าไปในระบบทันที



คลิกเมื่อต้องการเพิ่มรายชื่อ->

คลิกเมื่อต้องการลบข้อมูล->

เพิ่มรายชื่อ

ลบรายชื่อ

บันทึก

ปิดฟอร์ม

หมายเลขประจำตัวผู้ป่วย

401091001

ตำแหน่ง ชื่อ นามสกุล เพศ วัน/เดือน/ปีเกิด:

นาย กิตติพงษ์ เจริญผล ชาย

อายุ สถานภาพ สัญชาติ ศาสนา

21 โสด

บัตรประชาชน/บัตรข้าราชการเลขที่ สิทธิการรักษาพยาบาล

ที่อยู่ทะเบียนบ้าน

รหัสไปรษณีย์ โทรศัพท์:

ที่อยู่ปัจจุบัน

รหัสไปรษณีย์ โทรศัพท์:

อาชีพ วุฒินิติศึกษา

วันนี้วันที่

19 กุมภาพันธ์ 2544

เวลาตอนนี้

11:28:32

ก)

สถานที่ทำงาน โทรศัพท์:

ชื่อบิดา ชื่อมารดา ชื่อผู้สมรส

ชื่อผู้ติดต่อได้กรณีฉุกเฉิน เกี่ยวข้องเป็น

ที่อยู่ผู้ติดต่อ รหัสไปรษณีย์ เบอร์โทร:

แพทย์ โรคประจำตัว หมู่เลือด

เลขที่บัตรประกันสังคม บริษัท

วันที่ขอบัตร วันที่บัตรหมดอายุ ประกันกับโรงพยาบาล

ข)

รูปที่ 4.2 หน้าจอการเพิ่มประวัติผู้ป่วยใหม่ ก) ส่วนที่ 1 ข) ส่วนที่ 2

4.1.3 ฟอรั่มการแก้ไขประวัติผู้ป่วย

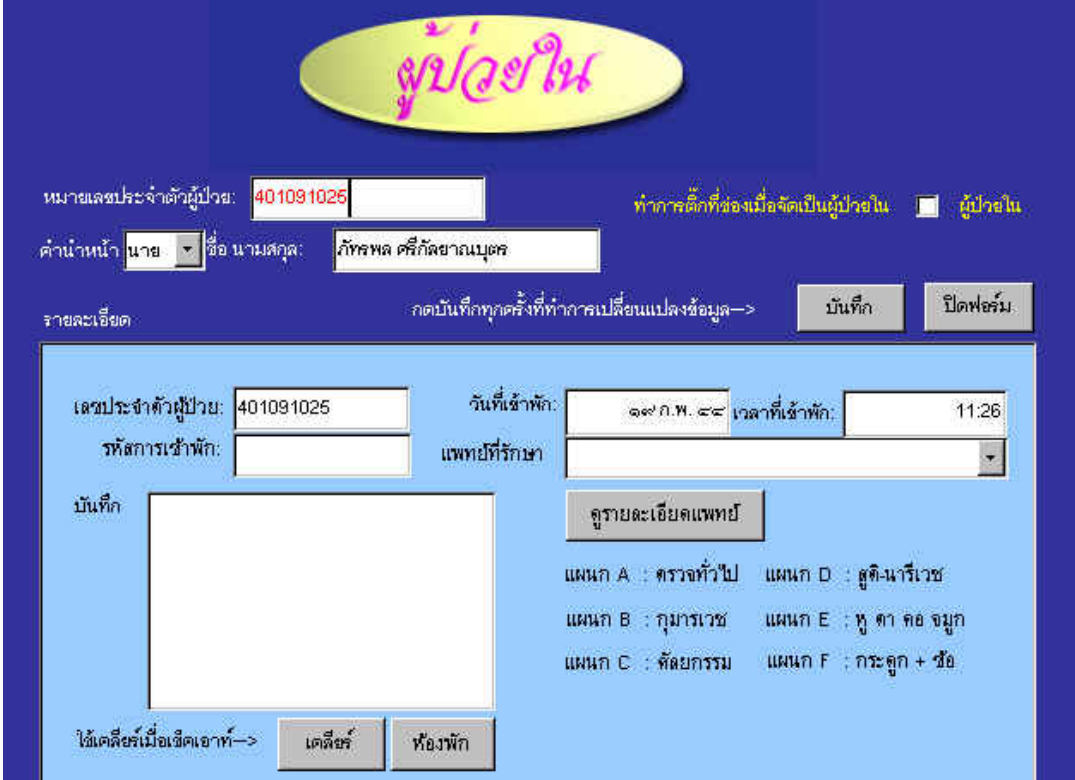
เมื่อทำการกดปุ่มแล้วจะมีให้กรอกหมายเลขประจำตัวผู้ป่วยโดยสามารถอ่านรหัสจากบัตรบาร์โค้ดหรือพิมพ์ลงไปก็ได้



รูปที่ 4.3 หน้าจอให้กรอกหมายเลขประจำตัวผู้ป่วย

4.1.4 ฟอรั่มการจัดคนไข้เข้าเป็นผู้ป่วยใน

เมื่อทำการกดปุ่มแล้วจะขึ้นหน้าจอให้ขึ้นเหมือนรูปที่ 4.3 เพื่อทำการใส่ค่าเข้าไปจากนั้นก็จะมีหน้าจอดังภาพที่ 4.4 เพื่อทำการบันทึกข้อมูลที่คนไข้ได้ทำการเลือกไว้เช่น รหัสการเข้าพักประเภทของห้องพัก และจะมีข้อมูลขึ้นให้เองแล้วจึงทำการบันทึกค่า ข้อมูลก็จะถูกเก็บเข้าไปในฐานข้อมูล และเมื่อจะทำการเช็คเอาท์ (Check Out) คือจัดผู้ป่วยออกจากโรงพยาบาลก็เข้ามาที่หน้าจอเดียวและทำการยืนยันหมายเลขประจำตัวผู้ป่วยแล้วกดบันทึกค่าเข้าไปก็จะเป็นอันเสร็จจากการเช็คเอาท์ของคนไข้



รูปที่ 4.4 หน้าจอผู้ป่วยในที่ใช้จัดคนไข้เข้า / ออก จากโรงพยาบาล

4.1.5 ฟอรั่มแสดงและแก้ไข ข้อมูลของผู้ป่วย

ในฟอรั่มนี้เมื่อเข้ามาแล้วจะเป็นการแสดงประวัติของผู้ป่วยจะใช้ในกรณีที่เป็นผู้ป่วยเก่ากลับเข้ามาทำการรักษาเจ้าหน้าที่ก็จะทำเพียงแค่นำบัตรประจำตัวผู้ป่วยมาอ่านบาร์โค้ดก็จะแสดงรายละเอียดทั้งหมดของคนไข้จากนั้นเมื่อเข้ามาในฟอรั่มนี้ก็จะจะมีปุ่มคำสั่งต่างๆ เช่นแก้ไขประวัติพิมพ์ประวัติ พิมพ์บัตรคนไข้ และการจัดลำดับบัตรคิว แสดงได้ดังรูปที่ 4.5

ค้นหาจากชื่อ-สกุล	แก้ไขข้อมูล	เพิ่มผู้ป่วยใหม่	บัตรคิว	ปิดฟอร์ม
ค้นหาจากเลขประจำตัว	ตัวอย่างก่อนพิมพ์	พิมพ์ประวัติ	พิมพ์บัตร	

ประวัติ

หมายเลขประจำตัวผู้ป่วย: 401091025

ตำแหน่ง: นาย ชื่อ นามสกุล: ภัทรพล ศรีภักขานมบุตร หมู่เลือด: O

แพ้ยา: โรคประจำตัว:

เพศ: ชาย วันเดือนปีเกิด: ๒๒ ก.พ. ๒๕๑๖ อายุ: 21 ปี

สัญชาติ: ไทย ศาสนา: พุทธ สถานภาพ: โสด

บัตรประชาชนฉบับราชการเลขที่: 3-3633-34109-12-1

ชื่อบิดา: อนิสร ชื่อมารดา: สุพรรณา

ชื่อผู้สมัคร:

ที่ติดต่อ

ที่อยู่ทะเบียนบ้าน: 1081 ซ. 7 มุขมนตรี รหัสไปรษณีย์: 30000 โทรศัพท์: (044)245-172

ที่อยู่ปัจจุบัน: 927/12 ลาดพร้าว รหัสไปรษณีย์: 10230 โทรศัพท์: (01)875-0432

สถานที่ทำงาน: โทรศัพท์:

ฉุกเฉิน

ชื่อผู้ติดต่อได้กรณีฉุกเฉิน: กร เกี่ยวข้องเป็น: พี่ชาย

ที่อยู่ผู้ติดต่อ: รหัสไปรษณีย์:

เบอร์โทร: (01) 827-6641

รูปที่ 4.5 หน้าจอแสดงรายละเอียดของผู้ป่วย

ซึ่งเมื่อทำการกดปุ่มเลือกแก้ไขข้อมูล โปรแกรมจะเปลี่ยนหน้าจอโดยมีรายละเอียดเหมือนกับการเพิ่มประวัติผู้ป่วย โดยจะแสดงดังภาพรูปที่ 4.6

คลิกเมื่อต้องการลบทั้งหมด→

ลบเรคคอร์ด

บันทึกใหม่

ปิดฟอร์ม

หมายเลขประจำตัวผู้ป่วย
401091025

คำนำหน้า ชื่อ นามสกุล เพศ วันเดือนปีเกิด:
นาย ภัทรพล ศรีภักดีชานบุตร ชาย ๒๒ ก.พ. ๒๕๑

อายุ สถานภาพ สัญชาติ ศาสนา
21 โสด ไทย พุทธ

บัตรประชาชน/บัตรข้าราชการเลขที่ สิทธิคำรักษาพยาบาล
3-3533-34 109-12-1 ชำระเอง

ที่อยู่ในทะเบียนบ้าน
1081 ซ. 7 มุขมนตรี
จังหวัดปทุมธานี โทรศัพท์:
30000 (044)245-172


ที่อยู่ปัจจุบัน
927/12 ลาดพร้าว
จังหวัดปทุมธานี โทรศัพท์:
10230 (01)875-0432

อาชีพ วุฒิการศึกษา
ปริญญาตรี

รูปที่ 4.6 หน้าจอการแก้ไขประวัติคนไข้

ในฟอร์มแสดงประวัติของผู้ป่วยนี้ยังสามารถพิมพ์ประวัติของคนไข้ออกมาดูได้ด้วยโดยสามารถกดปุ่มคำสั่งตัวอย่างก่อนพิมพ์แสดงดังรูปที่ 4.7

เมนู เพิ่ม แก้ไข ค้นหา พิมพ์



ศูนย์การแพทย์สมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี
คณะแพทยศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ประวัติผู้ป่วย

หมายเลขประจำตัวผู้ป่วย: 401091025

ชื่อ นาย ภัทรพล ศรีภักดียาณบุตร เพศ ชาย วัน/เดือน/ปีเกิด: ๒๒ ก.พ. ๒๑ อายุ: 21 ปี
 สถานภาพ โสด สัญชาติ: ไทย ศาสนา: พุทธ
 บัตรประชาชน/บัตรข้าราชการเลขที่: 3-3633-34 109-12-1
 สิทธิการรักษาพยาบาล ชำระเอง

ที่อยู่ทะเบียนบ้าน: 1081 ซ. 7 มุขมนตรี รหัสไปรษณีย์: 30000 โทรศัพท์: (044)245-172	ที่อยู่ปัจจุบัน: 927/12 ลาดพร้าว รหัสไปรษณีย์: 10230 โทรศัพท์: (01)875-0432
---	---

รูปที่ 4.7 ตัวอย่างก่อนพิมพ์ประวัติผู้ป่วย

ถ้าทำการเลือกคำสั่งพิมพ์บัตรคนไข้โปรแกรมก็จะพิมพ์บัตรคนไข้ออกมาและถ้าทำการเลือกพิมพ์บัตรคิว โปรแกรมก็จะพิมพ์บัตรคิวตามลำดับที่อยู่ถัดไปรูปที่ 4.8 แสดงลักษณะบัตรประจำตัวผู้ป่วย



รูปที่ 4.8 ลักษณะบัตรประจำตัวผู้ป่วย

ในกรณีที่ป้อนรหัสผู้ป่วยเข้าไปไม่ถูกต้องหรือไม่มีข้อมูลจะเกิดการแสดงผลหน้าจอ ดังรูปที่ 4.9



รูปที่ 4.9 หน้าจอแสดงผลเมื่อไม่มีข้อมูลผู้ป่วยในระบบ

4.1.6 ฟอรัมแสดงรายละเอียดของแพทย์

เมื่อทำการเลือกคำสั่งบนเมนูบาร์ และเลือกค้นหา และเลือกแพทย์ จะมีหน้าจอให้กรอกรหัสแพทย์ เมื่อทำการกรอกรหัสแพทย์ที่ต้องการก็จะแสดงผลหน้าจอ ดังรูปที่ 4.10

ข้อมูลแพทย์

รหัสประจำตัว:	A001
รหัสแผนก:	A
นาม:	นายแพทย์คมสัน ประเสริฐยิ่ง
ตำแหน่ง:	จักษุแพทย์
โทรศัพท์ติดต่อ:	01-8986454

แผนก A : ตรวจทั่วไป

แผนก B : กุมารเวช

แผนก C : ศัลยกรรม

แผนก D : สูติ-นารีเวช

แผนก E : บุตา คอ จมูก

แผนก F : กระดูก + ข้อ

ปิดฟอร์ม

รูปที่ 4.10 หน้าจอแสดงรายละเอียดของแพทย์

4.1.7 การกลับเข้าสู่เมนูหลัก

การกลับเข้าสู่เมนูหลักสามารถทำได้ โดยเลื่อนลูกศรเมาส์ (Mouse Pointer) ไปที่เมนู ดังแสดงในรูปที่ 4.11



รูปที่ 4.11 เมนูบาร์ (Menu Bar) ซึ่งสามารถกลับไปหน้าเมนูหลักได้

ถ้าทำการเลือกดูรายชื่อผู้ป่วย หน้าจอจะแสดงรายชื่อผู้ป่วยทั้งหมดที่อยู่ในฐานข้อมูลดังรูปที่ 4.12

หมายเลขประจำตัวผู้ป่วย	ชื่อ นามสกุล	วัน/เดือน/ปีเกิด	อายุ
401091001	กิตติพงศ์ เจริญผล	๒๘ ต.ค. ๒๒	21
401091002	เกรียงไกร คำสงค์	๒๐ ต.ค. ๒๑	22
401091005	เฉลิมพล วงศ์ทยานิธิ	๕ ก.ค. ๒๐	23
401091006	ชาญเดช พรหมมณี	๑๒ มี.ย. ๒๑	22
401091007	ชำนาญพงศ์ อาสน์ทอง	๗ พ.ค. ๒๒	21
401091009	ฐิติรัตน์ คุ่มสุพรรณ	๒๙ ต.ค. ๒๒	21
401091010	ณัฐณี วิทยายุทธ	๑๓ ธ.ค. ๒๑	22
401091011	เทพรัตน์ ลายลักษณ์	๖ มี.ค. ๒๒	21
401091012	เทียนชัย กวรมรัมย์	๒๖ ก.พ. ๒๐	23
401091013	ธีรยุทธ ล่วงศรี	๔ ก.พ. ๒๒	21
401091014	นาวิน โฉมฉาย	๑๑ เม.ย. ๒๑	22
401091015	นิพนธ์ เจนพิชิตกุลชัย	๑๑ ก.ค. ๒๑	22

รูปที่ 4.12 หน้าจอแสดงตารางรายชื่อของผู้ป่วยในโรงพยาบาล

4.2 ผลการทดลองของรถยนต์อัตโนมัติ

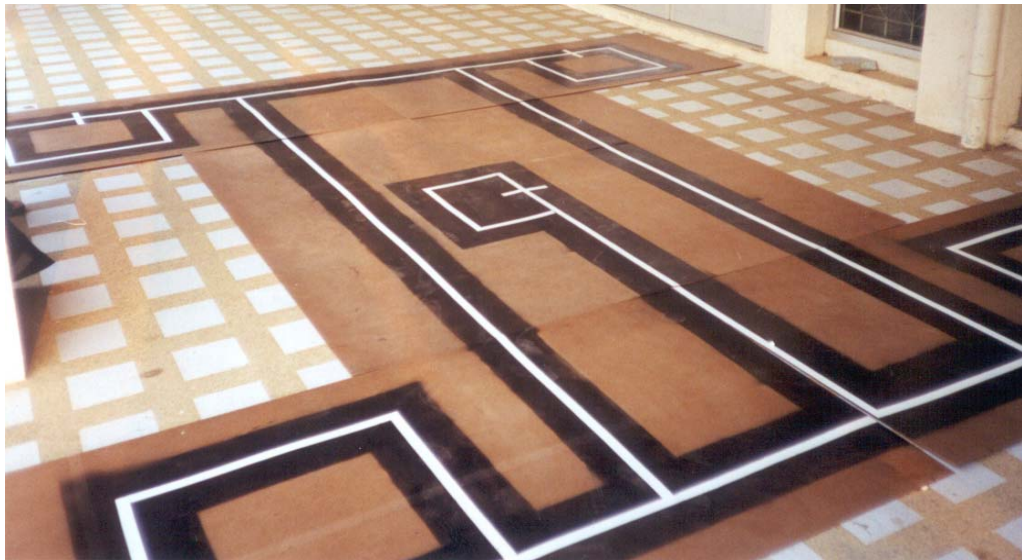
4.2.1 วงจรขับเคลื่อนไฟฟ้ากระแสตรง

วงจรขับเคลื่อนไฟฟ้ากระแสตรงสามารถวัดแรงดันเอาต์พุต (Output) ที่ป้อนให้กับมอเตอร์มีค่าเท่ากับ 24 โวลต์

4.2.2 วงจรตรวจจับเส้นทาง

วงจรตรวจจับเส้นทางอาศัยการสะท้อนแสงจากสีของพื้นที่แตกต่างกันอย่างชัดเจนโดยในการทดลองนี้ใช้พื้นสีดำและเส้นทางเดินของรถยนต์อัตโนมัติเป็นสีขาว แสดงได้ดังรูปที่ 4.13 และแรงดันเอาต์พุตจากวงจรตรวจจับเส้นทางเมื่อตรวจจับได้พื้นสีขาวมีค่าเท่ากับ 3.4 โวลต์ซึ่งถือว่าเป็นลอจิก 1 ส่วนแรงดันเอาต์พุตจากวงจรตรวจจับเส้นทางเมื่อตรวจจับได้พื้นสีดำมีค่าเท่ากับ 0.8

โวลท์ ซึ่งถือได้ว่าเป็นลอจิก 0 ระยะห่างระหว่างพื้นกับเซนเซอร์เท่ากับ 1.5 เซนติเมตร ความกว้างของแถบสีขาวซึ่งเป็นทางเดินของรถขนส่งอัตโนมัติมีความกว้าง 3.0 เซนติเมตร



รูปที่ 4.13 เส้นทางเดินของรถขนส่งอัตโนมัติ

4.2.3 รถขนส่งอัตโนมัติ

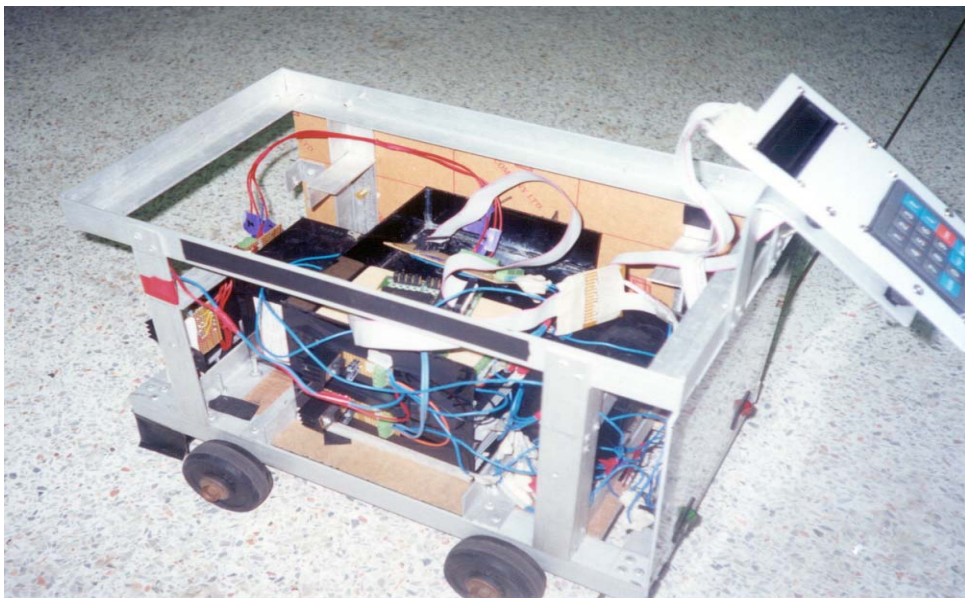
4.2.3.1 รถขนส่งอัตโนมัติจะเริ่มทำงานเมื่อ เจ้าหน้าที่ทำการป้อนคำสั่งที่เป็นตัวเลขของแผนกต้นทางและแผนกปลายทางผ่านทางแป้นพิมพ์ (Keypad) ที่อยู่บนตัวรถขนส่งอัตโนมัติ รถขนส่งอัตโนมัติจะวิ่งไปจนถึงสถานีปลายทางแล้วจะหยุด จนกว่าจะมีเจ้าหน้าที่มาป้อนคำสั่งให้ทำงานอีกครั้ง โดยรูปที่ 4.14 แสดงแป้นพิมพ์และจอแสดงผลแบบแอลซีดี (LCD Display) ที่แสดงแผนกต้นทางและแผนกปลายทาง รูปที่ 4.15 แสดงวงจรภายในรถขนส่งอัตโนมัติ และรูปที่ 4.16 แสดงรถขนส่งอัตโนมัติขณะวิ่งอยู่บนเส้นทางที่กำหนด

4.2.3.2 น้ำหนักของตัวรถขนส่งอัตโนมัติประมาณ 7 กิโลกรัม

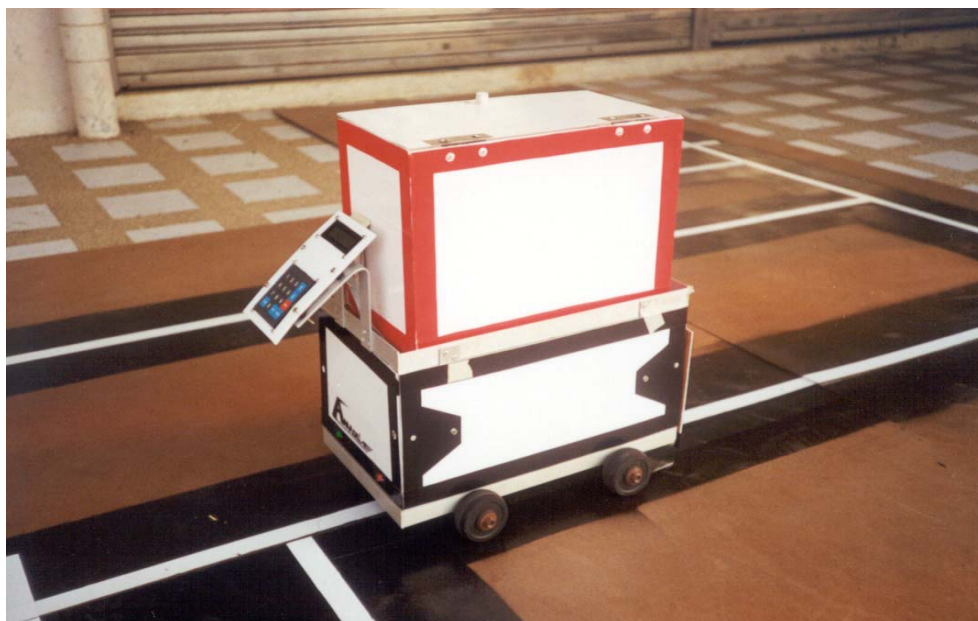
4.2.3.3 รถสามารถบรรทุกน้ำหนักได้ไม่เกิน 10 กิโลกรัม



รูปที่ 4.14 แสดงเป็นพิมพ์และจอแสดงผลแบบแอลซีดี (LCD Display) ที่แสดงแผนกั้นทาง และ แผนกปลายทาง



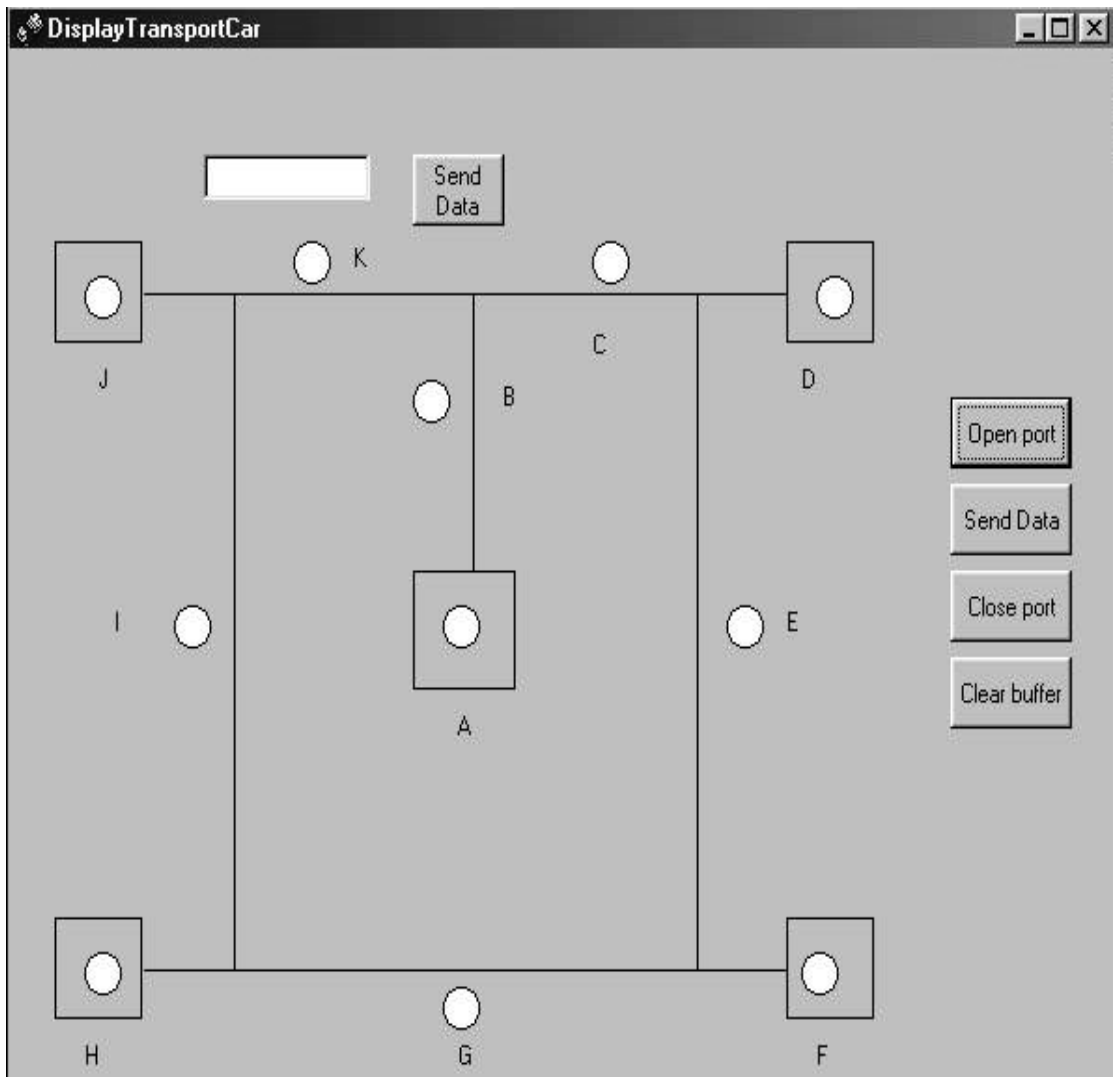
รูปที่ 4.15 แสดงวงจรภายในรถขนส่งอัตโนมัติ



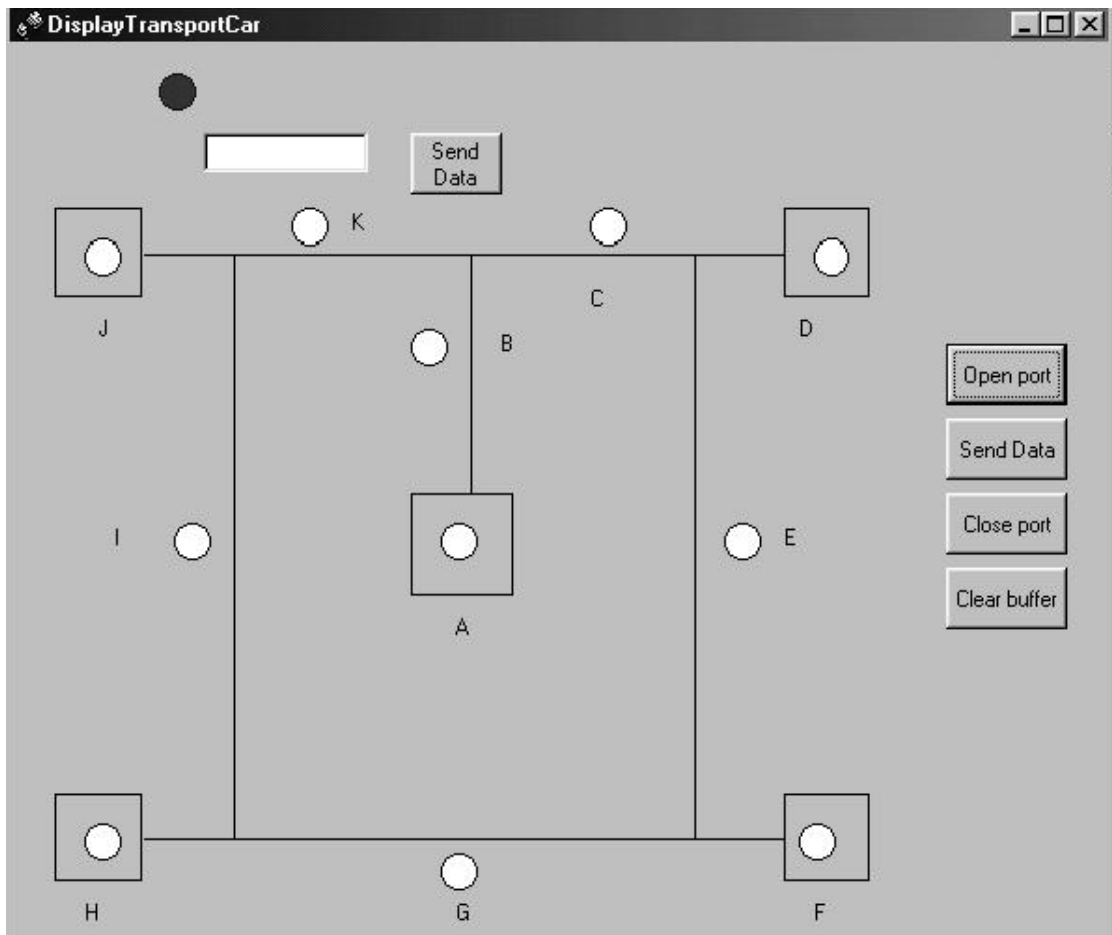
รูปที่ 4.16 รถขนส่งอัตโนมัติขณะวิ่งอยู่บนเส้นทางที่กำหนด

4.3 ผลการทดลองของระบบตรวจสอบตำแหน่งของรถขนส่งอัตโนมัติ

จากการทดลองตรวจสอบตำแหน่งของรถขนส่งอัตโนมัติ แล้วนำมาแสดงผลผ่านจอคอมพิวเตอร์ด้วยโปรแกรมวิชวลเบสิก (Visual Basic) แสดงได้ดังรูปที่ 4.16 ก) หน้าจอคอมพิวเตอร์เมื่อเปิดโปรแกรม และ ข) หน้าจอคอมพิวเตอร์เมื่อกดปุ่ม Open port เพื่อรับค่าจากเซนเซอร์ตรวจจับตำแหน่งของรถขนส่งอัตโนมัติ



n)



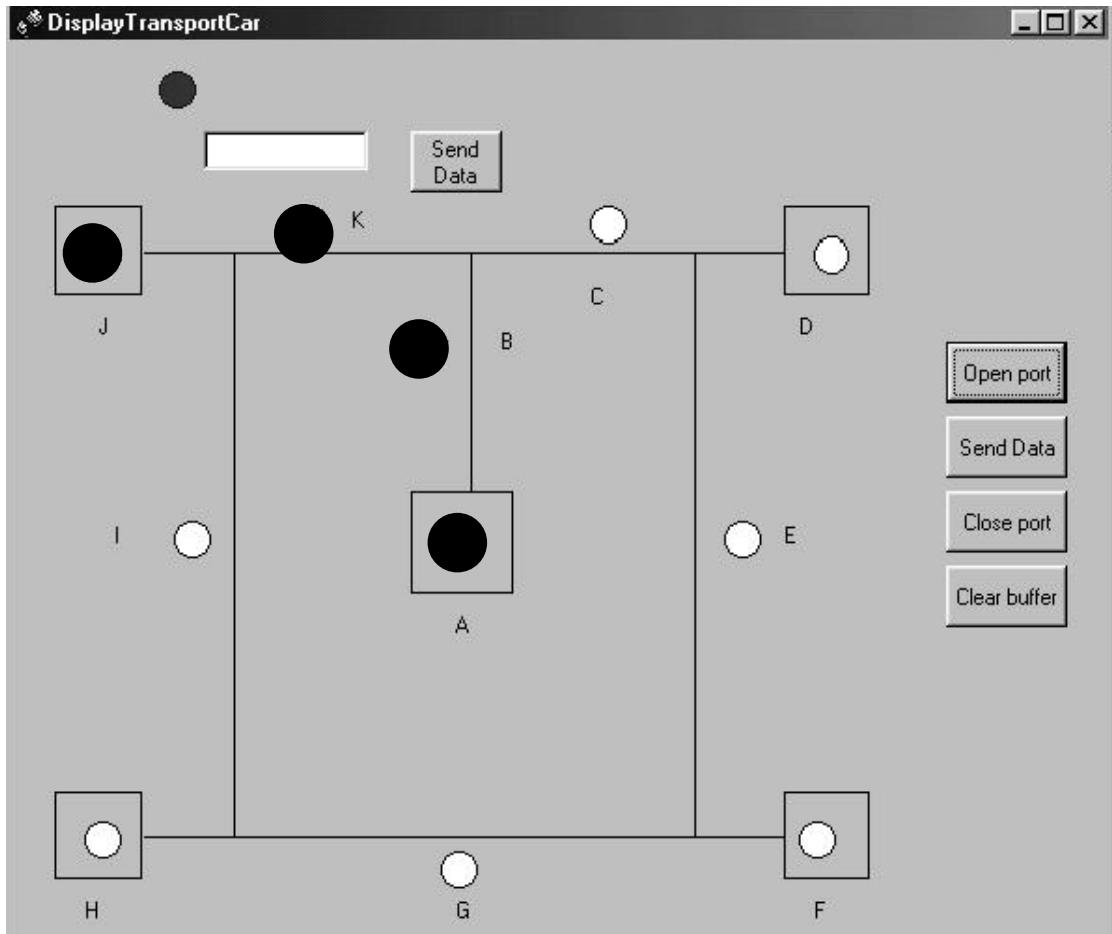
ข)

- สมมุติให้ ○ เป็นเซนเซอร์ตรวจจับเส้นทาง
 ตำแหน่ง A เป็นสถานี (แผนกทะเบียนประวัติ)
 ตำแหน่ง J เป็นแผนกแผนกกุมารเวช
 ตำแหน่ง D เป็นแผนกทันตกรรม
 ตำแหน่ง H เป็นแผนกจักษุเวช
 ตำแหน่ง F เป็นแผนกศัลยกรรม

รูปที่ 4.17 หน้าจอเมื่อเปิดโปรแกรมระบบตรวจสอบตำแหน่งของรถขนส่งอัตโนมัติ

4.3.1 ทดลองส่งรถขนส่งอัตโนมัติออกจากตำแหน่ง A ไปยังตำแหน่ง J

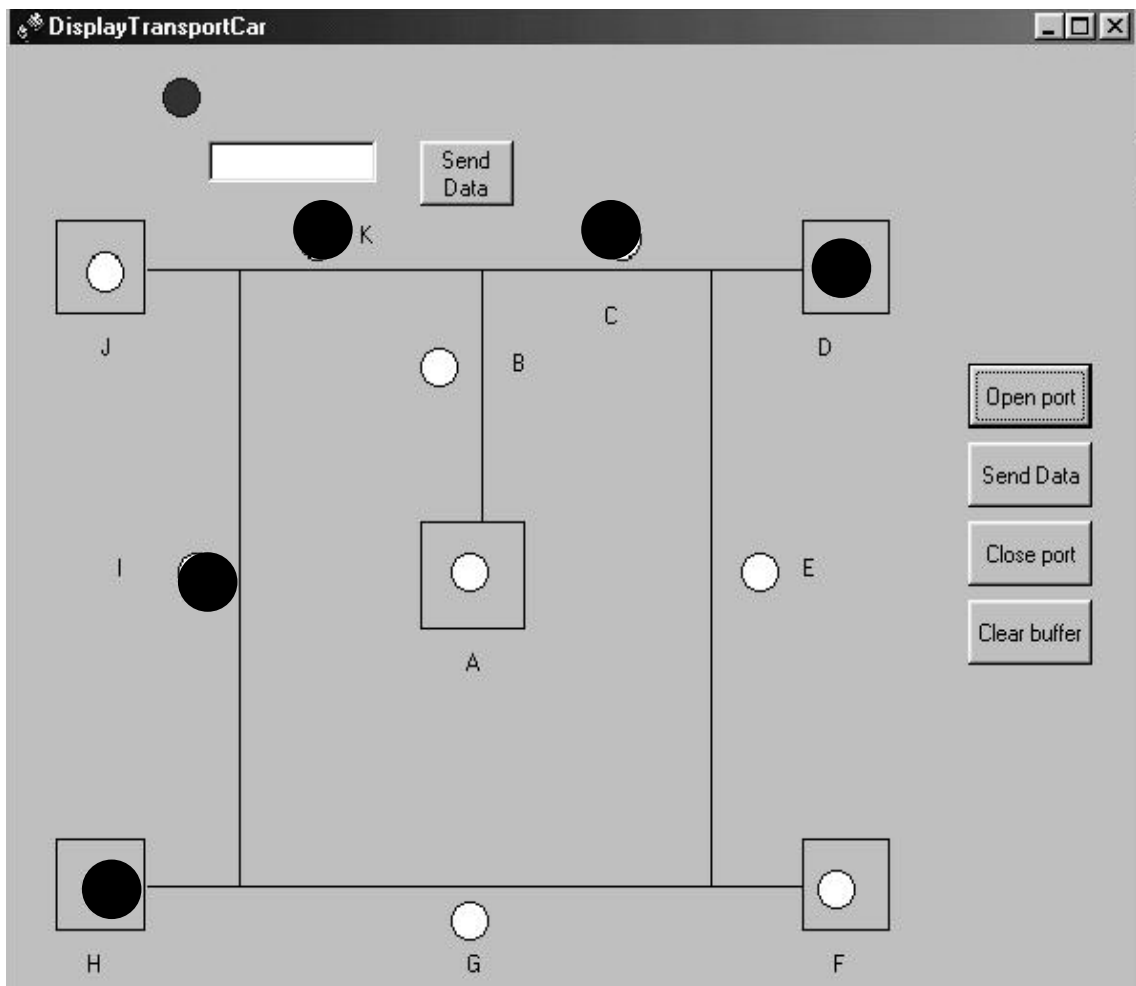
ผลที่แสดงบนหน้าจอคอมพิวเตอร์เมื่อรถขนส่งอัตโนมัติวิ่งจากตำแหน่ง A ไปยังตำแหน่ง J แสดงได้ดังรูปที่ 4.17



รูปที่ 4.18 หน้าจอคอมพิวเตอร์เมื่อรถขนส่งอัตโนมัติวิ่งจากตำแหน่ง A ไปยังตำแหน่ง J

4.3.2 ทดลองส่งรถขนส่งอัตโนมัติออกจากตำแหน่ง D ไปยังตำแหน่ง H

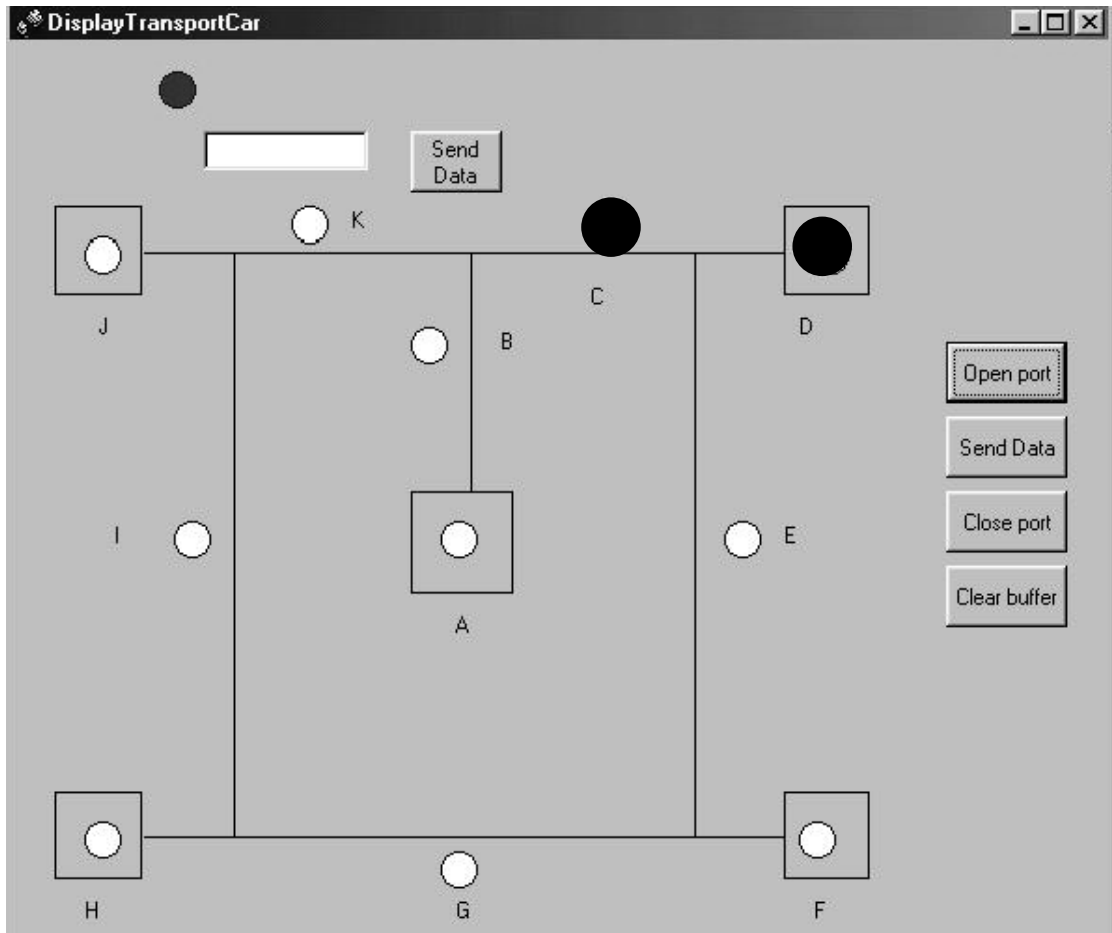
ผลที่แสดงบนหน้าจอคอมพิวเตอร์เมื่อรถขนส่งอัตโนมัติวิ่งจากตำแหน่ง D ไปยังตำแหน่ง H แสดงได้ดังรูปที่ 4.18



รูปที่ 4.19 หน้าจอคอมพิวเตอร์เมื่อรถขนส่งอัตโนมัติวิ่งจากตำแหน่ง D ไปยังตำแหน่ง H

4.3.2 ทดลองส่งรถขนส่งอัตโนมัติออกจากตำแหน่ง D ไปยังตำแหน่ง H

ผลที่แสดงบนหน้าจอคอมพิวเตอร์เมื่อรถขนส่งอัตโนมัติวิ่งจากตำแหน่ง D แต่ไปไม่ถึงตำแหน่ง H เนื่องจากรถเกิดหยุดอยู่ระหว่างตำแหน่ง K และ C แสดงได้ดังรูปที่ 4.19 ทำให้เจ้าหน้าที่ทราบตำแหน่งที่รถขนส่งอัตโนมัติหยุดทำงาน หรือ ออกนอกเส้นทาง



รูปที่ 4.20 หน้าจอคอมพิวเตอร์เมื่อส่งรถขนส่งอัตโนมัติจากตำแหน่ง D ไปตำแหน่ง H แต่เกิด ความผิดพลาดในการทำงานที่ระหว่างตำแหน่ง K และ C

บทที่ 5

สรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

5.1.1 สรุปผลการดำเนินงานและปัญหาการทดลองใน โปรแกรมจัดการฐานข้อมูลด้วยระบบบาร์โค้ด

จากการทดลองใน โปรแกรมจัดการฐานข้อมูลด้วยระบบบาร์โค้ดพบว่า โปรแกรมจัดการฐานข้อมูลด้วยระบบบาร์โค้ดกับเครื่องอ่านแถบรหัสสามารถทำงาน ได้อย่างดี แต่ในบางรหัสมีขนาดเล็กมากๆ เครื่องจะไม่สามารถอ่านได้ ทั้งนี้เพราะจุดรวมแสง (Focus) ของแสงมีเส้นผ่านศูนย์กลางใหญ่กว่าความกว้างของแถบแคบ จึงทำโฟโต้ดีเทคเตอร์ไม่สามารถรับได้

5.1.1.1 สำหรับเครื่องอ่านแถบรหัส เมื่อนำมาใช้กับแถบรหัสของบัตรประจำตัวคนไข้ที่จัดทำขึ้นเป็นพิเศษจะเป็นรหัสเดียวกับรหัสประจำตัวนักศึกษาของนิสิตคณะวิศวกรรมศาสตร์ภาควิชาวิศวกรรมไฟฟ้าและมีแถบชนิดเดียวกับบัตรนิสิต พบว่าสามารถอ่านรหัสได้ทุกรหัสแต่ในที่นี้ใช้รหัสบาร์โค้ดแบบ 3 of 9 แต่ในการใช้ต้องทำมุมในการอ่านให้พอเหมาะประมาณ 20 องศาที่เป็นเช่นนี้เพราะรูที่หัวอ่านที่จะให้แสงออกมาตกกระทบกับรหัสและสะท้อนกลับเล็กน้อย ส่วนรหัสที่เล็กมากๆ ก็ไม่สามารถอ่านรหัสได้

5.1.1.2 ในการต่อเครื่องอ่านบาร์โค้ด เครื่องอ่านบาร์โค้ดจะประกอบด้วยสายที่ต่อกับคีย์บอร์ดและหัวอ่าน โดยพอร์ตที่ต่อนี้จะต่อที่พอร์ตของคีย์บอร์ดนั่นเอง เมื่อนำมาใช้งานกับเครื่องคอมพิวเตอร์และตัวโปรแกรมจะพบว่าเกิดความ ผิดพลาดในการแสดงผลของข้อมูลในการอ่าน บาร์โค้ดเนื่องจากโปรแกรมที่ใช้เป็นวินโดว (Windows 98) ในการอ่านบาร์โค้ดจึงต้องเลือกเมนูเป็นภาษาอังกฤษ (En) จึงจะทำงานได้ตามโปรแกรมที่ออกแบบไว้ อีกทั้งในการเพิ่มข้อมูลสามารถใช้ร่วมกับคีย์บอร์ดในการป้อน ข้อมูลซึ่งจะมีแบบฟอร์มของการกรอกแสดงไว้

5.1.1.3 ในส่วนของโปรแกรมการพิมพ์รหัสแถบสามารถพิมพ์รหัสแบบ 3 ใน 9 ได้แต่การพิมพ์ยังมีข้อผิดพลาดบ้างดังนี้

ก) ขนาดของการพิมพ์บางครั้งได้ไม่ตรงตามความต้องการ เช่น ต้องการพิมพ์รหัสแถบแบบ 3 ใน 9 แบบสอดคล้องมาตรฐานที่ความสูงตามความเหมาะสมของตัวงาน เพราะถ้าพิมพ์ในบัตรก็

ต้องใช้ความสูง 7 มิลลิเมตร ถ้าเป็นเครื่องมือใช้ความสูง 1.5 เซนติเมตร แต่ผลที่ออกมาได้ขนาดความสูง 2 ซม.

ข) รหัสแถบที่มีข้อมูลยาวๆ เมื่อพิมพ์ออกมารหัสจะขาดหายไปเล็กน้อยในบางแถบจึงทำให้ข้อมูลที่อ่านได้มีความคลาดเคลื่อน

5.1.1.4 โปรแกรมจัดการฐานข้อมูลด้วยระบบบาร์โค้ด เป็นการสร้างซอฟต์แวร์ที่มีการใช้งานที่สะดวกเองจากสร้างมาจากไมโครซอฟท์แอคเซสเวอร์ชัน 4 (Microsoft Access Version 4) และมีการใช้งานโดยการเชื่อมต่อกับเครื่องอ่านบาร์โค้ดแต่โปรแกรมนี้ยังมีปัญหาอยู่บ้าง ดังนี้

ก) ในการบันทึกข้อมูลนั้นจะต้องรอรยะเวลาหนึ่งก่อนที่จะสามารถรับข้อมูลใหม่มาได้ ทั้งนี้เนื่องจากการถ่ายเทข้อมูลนั้นใช้เวลานาน

ข) โปรแกรมที่ใช้เป็นโปรแกรมไมโครซอฟท์แอคเซสเวอร์ชัน 4 (Microsoft Access Version 4) ที่ใช้กับวินโดวส์ 95 (Windows 95) ขึ้นไป การแสดงข้อความบางอย่างผิดพลาดไป เช่น ในการอ่านบาร์โค้ดของบัตรคนไข้ ต้องกดที่เมนู ไทยเป็นอังกฤษ มิฉะนั้นเมื่ออ่าน บาร์โค้ด เข้าไป โปรแกรมจะแสดงรหัสเป็นอักษรไทยแทนที่จะเป็น เลขอารบิก เช่น 401091025 ถ้าใช้เป็นภาษาอังกฤษ สามารถที่ใช้ได้เลยโดยไม่ต้องเปลี่ยน เพราะเมนูขึ้นมาเป็นอังกฤษค่าที่อ่านจะเป็น ตัวเลขจากการที่ได้ศึกษาส่วนต่างๆ ในโครงการนี้จะพบปัญหาและวิธีแก้ปัญหามากมาย ซึ่งสามารถสรุปข้อเสนอแนะได้หลายข้อ และหวังว่าผู้ที่ทำการศึกษาโครงการนี้ต่อไปในอนาคตจะได้นำข้อเสนอแนะข้างล่างนี้ไปใช้ประกอบการศึกษาต่อไป

5.1.2 สรุปผลการดำเนินงานและปัญหาการทดลองของรถขนส่งอัตโนมัติ

5.1.2.1 ปัญหาในการประกอบโครงรถ

- ก) การออกแบบไม่ระบุขนาดที่แน่นอนในการออกแบบทำให้การประกอบล่าช้า
- ข) การประกอบโครงรถทำตามแบบไม่ได้ในบางจุด
- ค) บางส่วนของตัวรถมีความไม่แข็งแรงเนื่องจากความไม่ชำนาญของผู้ประกอบ

5.1.2.2 ปัญหาด้านอิเล็กทรอนิกส์

- ก) การต่อวงจรทำได้ไม่เรียบร้อยเท่าที่ควรทำให้วงจรมีเสถียรภาพต่ำ
- ข) การวางวงจรภายในตัวรถไม่ได้ออกแบบไว้อย่างแน่นอนทำให้ไม่เป็นระเบียบ
- ค) การติดแผ่นระบายความร้อน (Heat Sink) โดยไม่รองด้วยแผ่นไมก้าทำให้เกิดการลัดวงจรได้

ง) แรงดันเอาท์พุทจากคอนโทลเลอร์มีต่ำ จึงไม่สามารถไบแอส (Bias) ทรานซิสเตอร์ (Transistor) ได้โดยตรง

จ) วงจรตรวจจ่ายเส้นทางมีแสงรบกวนทำให้เกิดความผิดพลาดในการทำงาน

ฉ) แบตเตอรี่มีความจุน้อยทำให้ต้องชาร์จ (Charge) บ่อย

5.1.3 สรุปผลการดำเนินงานและปัญหาการทดลองของระบบตรวจสอบตำแหน่งของรถขนส่งอัตโนมัติ

5.1.3.1 ต้องใช้แหล่งจ่ายไฟให้กับวงจรเซนเซอร์หลายชุดเพราะเซนเซอร์แต่ละตัววางห่างกัน

5.1.3.2 มีแรงดันเอาท์พุท (Output) จากเซนเซอร์มาที่คอนโทลเลอร์ต่ำ จนบางครั้งเกิดการประมวลผลผิดพลาด

5.2 สรุปผลงานที่ได้รับ

ผลงานที่ได้จากการทำโครงการนี้ ได้แก่

5.2.1 การเก็บและแสดงประวัติของคนไข้แต่ละคน

5.2.2 การบันทึกข้อมูลของคนไข้ และการบันทึกเข้าเป็นผู้ป่วยใน

5.2.3 การพิมพ์บัตรประจำตัวคนไข้ บัตรคิว

5.2.4 รถขนส่งอัตโนมัติต้นแบบ

5.3 ข้อเสนอแนะ

5.3.1 การสร้างฐานข้อมูล และ แอปพลิเคชัน (Application) ที่ ไมโครซอฟท์ แอ็กเซสเวอร์ชัน 4 (Microsoft Access Version 4) เป็นฐานข้อมูลที่มีความเสี่ยงค่อนข้างสูง เนื่องจากทั้งฐานข้อมูล, ฟอร์ม, คิวรี, รายงาน และ โปรแกรมที่ได้จัดทำขึ้นได้มีการเก็บไว้ในไฟล์เดียวกันทั้งหมด ถ้าหากไฟล์นั้นเกิดเสียหายอาจทำให้ข้อมูลสูญหายได้ ดังนั้นเพื่อเป็นการเสียดังกล่าว เราควรจะมีการคัดลอกไฟล์ไว้เป็นข้อมูลสำรองเสมอ ๆ หลังจากการใช้งานทุกครั้ง

5.3.2 การทำการสำรองข้อมูลในไมโครซอฟท์ แอ็กเซสเวอร์ชัน 4 (Microsoft Access Version 4) สามารถทำได้โดยการคัดลอกไฟล์ .MDB ไปไว้บนแผ่นดิสก์อื่นที่ไม่ได้ถูกใช้งาน โดยอาจทำการจัดเก็บแบบ บีบอัด (Compress) เพื่อลดเนื้อที่ในการจัดเก็บข้อมูลก็ได้

5.3.3 การทำงานของโปรแกรมไมโครซอฟท์แอคเซสเวอร์ชัน 4 (Microsoft Access Version) ต้องการทรัพยากรที่มีประสิทธิภาพค่อนข้างสูง มิฉะนั้นการทำงานของโปรแกรมจะช้ามาก

5.3.4 ควรจะศึกษาเกี่ยวกับเครื่องอ่านแถบรหัสที่ใช้เลเซอร์ไดโอด (Laser Diode) ในการเป็นแหล่งกำเนิดแสงและค้นหาตัวรับที่มีความไวและมีประสิทธิภาพมาก ๆ

5.3.5 โปรแกรมจัดการฐานข้อมูลด้วยระบบบาร์โค้ด สามารถทำไปประยุกต์แก้ไขและเพิ่มเติม จัดสร้างเป็นโปรแกรมเพื่อนำมาใช้ เช่น การจ่ายยา การคิดค่าบริการ เป็นต้น

5.3.6 ปัจจุบันได้มีการนำรหัสแถบชนิด UPC และ EAN มาเป็นรหัสในสินค้าหลายชนิดตามห้างสรรพสินค้าเพื่อแทนราคาหรือจำนวนสินค้าในคลัง ดังนั้นจึงควรศึกษาข้อมูลของรหัสทั้งสองแบบเพื่อนำมาปรับปรุงโปรแกรมพิมพ์รหัสแถบแบบ UPC หรือ EAN ได้

5.3.7 ควรใช้แหล่งจ่ายไฟแบบอื่นที่ไม่ใช่แบตเตอรี่ให้ตัวรถขนส่งเพื่อความสะดวกและใช้งานได้อย่างต่อเนื่อง

5.3.8 ควรใช้ระบบตรวจจับเส้นทางที่มีความแม่นยำมากขึ้น

5.3.9 ควรออกแบบโครงสร้างภายนอกของรถให้เหมาะสมกับงาน

5.3.10 ระบบตรวจสอบตำแหน่งรถควรมีเซนเซอร์ที่เหมาะสมมากขึ้น

การออกแบบระบบฐานข้อมูลด้วยโปรแกรมไมโครซอฟต์แอกเซส 97 (Microsoft Access 97) จะมีเริ่มต้นด้วยการออกแบบตาราง (Table) เพื่อนำมาสร้างความสัมพันธ์จากนั้นก็ให้นำตารางนั้นไปสร้างรูปแบบด้วยฟอร์ม (Form) เพื่อให้ง่ายต่อการใช้งานและในบางกรณีเมื่อต้องการเขียนลักษณะพิเศษของฐานข้อมูลจำเป็นจะต้องทำการเขียนคลาสโมดูล (Class Module) และแมโคร (Macro) เพื่อให้ได้ระบบตามที่ต้องการซึ่งในโปรแกรมของโครงการนี้ ได้ทำการออกแบบส่วนต่างๆไว้ ดังต่อไปนี้

1. ตาราง (Table)

1.1. ตาราง Detailnotecomein

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	ดัชนี	หมายเหตุ
codecomein	Text	5	ใช่ (มีค่าซ้ำไม่ได้)	คีย์หลัก
numberroom	Text	50	ใช่ (มีค่าซ้ำได้)	

1.2 ตาราง Doctor

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	ดัชนี	หมายเหตุ
iddoctor	Text	4	ใช่ (มีค่าซ้ำไม่ได้)	คีย์หลัก
codestation	Text	1	ใช่ (มีค่าซ้ำได้)	
Doctor	Text	50		
Day	Text	50		
tel	Text	10		

1.3 ตาราง Intro

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	ดัชนี	หมายเหตุ
key	Text	1	ใช่ (มีค่าซ้ำไม่ได้)	คีย์หลัก
intro	Text	10		

1.4 ตาราง Married

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	ดัชนี	หมายเหตุ
idmarried	Text	2	ใช่ (มีค่าซ้ำไม่ได้)	คีย์หลัก
married	Text	10		

1.5 ตาราง Notecomein

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	ดัชนี	หมายเหตุ
codecomein	Text	5	ใช่ (มีค่าซ้ำไม่ได้)	คีย์หลัก
code	Text	9	ใช่ (มีค่าซ้ำได้)	
datecomein	Date/Time			Date ()
timein	Date/Time			Time ()
iddoctor	Text	10	ใช่ (มีค่าซ้ำได้)	
note	Text	100		

1.6 ตาราง Numberroom

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	ดัชนี	หมายเหตุ
numberroom	Text	5	ใช่ (มีค่าซ้ำไม่ได้)	คีย์หลัก
coderoom	Number	Long Integer	ใช่ (มีค่าซ้ำได้)	
floor	Number	Long Integer		

1.7 ตาราง Patient

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	ดัชนี	หมายเหตุ
code	Text	9	ใช่ (มีค่าซ้ำไม่ได้)	คีย์หลัก
key	Text	1	ใช่ (มีค่าซ้ำได้)	
name	Text	50	ใช่ (มีค่าซ้ำไม่ได้)	
Type	Text	1	ใช่ (มีค่าซ้ำได้)	
birthday	Date /			

	Time			
old	Text	3		
idmarried	Text	2	ใช่ (มีค่าใช่ได้)	
sachat	Text	15		

1.7 ตาราง Patient (ต่อ)

areligion	Text	10		
idcard	Text	20	ใช่ (มีค่าใช่ไม่ได้)	
codestatus	Text	1	ใช่ (มีค่าใช่ได้)	
address1	Text	60		
post1	Text	5	ใช่ (มีค่าใช่ได้)	
telephon1	Text	30		
address2	Text	60		
post2	Text	5		
telephone2	Text	30		
work	Text	30		
education	Text	30		
workstation	Memo			
tel	Text	30		
father	Text	50		
mother	Text	50		
fan	Text	50		
ERname	Text	50		
Relationship	Text	15		
address3	Text	60		
post3	Text	5		
telephone3	Text	15		
antidrug	Text	50		
ability	Text	50		
group	Text	10		

numbersocial	Text	12	ใช่ (มีค่าซ้ำไม่ได้)	
liberty	Text	50		
date	Date/Time			
exp	Date/Time			
hospital	Text	40		
in-out	Yes/No			Yes / No

1.8 ตาราง PickComein

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	คั้งนี้	หมายเหตุ
Count	Autonumber	Long Integer	ใช่ (มีค่าซ้ำไม่ได้)	คั้งหลัก
codecomein	Text	5	ใช่ (มีค่าซ้ำไม่ได้)	
code	Text	9	ใช่ (มีค่าซ้ำไม่ได้)	
datecomein	Date/Time			Date()
timein	Date/Time			Time()

1.9 ตาราง PickOut

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	คั้งนี้	หมายเหตุ
Count	Autonumber	Long Integer	ใช่ (มีค่าซ้ำไม่ได้)	คั้งหลัก
codecomein	Text	5	ใช่ (มีค่าซ้ำไม่ได้)	
code	Text	9	ใช่ (มีค่าซ้ำไม่ได้)	
dateout	Date/Time			Date()
timeout	Date/Time			Time()

1.10 ตาราง Q

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	คั้งนี้	หมายเหตุ
Q	Autonumber	Long Integer	ใช่ (มีค่าซ้ำไม่ได้)	คั้งหลัก
code	Text	9	ใช่ (มีค่าซ้ำไม่ได้)	
timecome	Date/Time			Time()
roomheal	Text	3		
cause	Text	50		

1.11 ตาราง Room

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	ดัชนี	หมายเหตุ
coderoom	Number	Long Integer	ใช่ (มีค่าซ้ำไม่ได้)	คีย์หลัก
room	Text	50		
detailroom	Text	100		

1.12 ตาราง Roomheal

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	ดัชนี	หมายเหตุ
roomheal	Text	3	ใช่ (มีค่าซ้ำไม่ได้)	คีย์หลัก
iddoctor	Text	4		
now	Yes/No			

1.13 ตาราง Station

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	ดัชนี	หมายเหตุ
codestation	Text	1	ใช่ (มีค่าซ้ำไม่ได้)	คีย์หลัก
station	Text	25		
floorstation	Text	2		

1.14 ตาราง Status

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	ดัชนี	หมายเหตุ
codestatus	Text	1	ใช่ (มีค่าซ้ำไม่ได้)	คีย์หลัก
typestatus	Text	20		

1.15 ตาราง Type

ชื่อเขตข้อมูล	ชนิด	ขนาดเขตข้อมูล	ดัชนี	หมายเหตุ
Type	Autonumber	Long Integer	ใช่ (มีค่าซ้ำไม่ได้)	คีย์หลัก
sex	Text	5	ใช่ (มีค่าซ้ำกันได้)	

2. แบบสอบถาม (Query)

2.1 แบบสอบถาม AllComein

เขตข้อมูล	ตาราง	แสดง
Count	PickComein	ใช่
code	PickComein	ใช่
datecomein	PickComein	ใช่
timein	PickComein	ใช่
dateout	PickOut	ใช่
timeout	PickOut	ใช่

2.2 แบบสอบถาม AllPatient

เขตข้อมูล	ตาราง	แสดง
Code	Patient	ใช่
name	Patient	ใช่
birthday	Patient	ใช่
old	Patient	ใช่

2.3 แบบสอบถาม AllQ

เขตข้อมูล	ตาราง	แสดง
Q	Q	ใช่
code	Patient	ใช่

name	Patient	ใช่
timecome	Patient	ใช่
cause	Q	ใช่

2.4 แบบสอบถาม ComeinNow

เขตข้อมูล	ตาราง	แสดง
codecomein	notecomein	ใช่
code	Patient	ใช่
name	Patient	ใช่
datecomein	notecomein	ใช่
timein	notecomein	ใช่
doctor	doctor	ใช่
note	notecomein	ใช่

2.5 แบบสอบถาม Doctors

เขตข้อมูล	ตาราง	แสดง
iddoctor	doctor	ใช่
doctor	doctor	ใช่
station	station	ใช่
day	doctor	ใช่
tel	doctor	ใช่

2.6 แบบสอบถาม SearchComein

เขตข้อมูล	ตาราง	แสดง
codecomein	notecomein	ใช่

name	Patient	ใช่
datecomein	notecomein	ใช่
doctor	doctor	ใช่
numberroom	detailnotecomein	ใช่

3. กลาสโมดูล(Code)

3.1 กลาสโมดูลของ Form AddDoctors

```
Private Sub Command7_Click()
```

```
On Error GoTo Err_Command7_Click
```

```
DoCmd.GoToRecord , , acNewRec
```

```
Exit_Command7_Click:
```

```
Exit Sub
```

```
Err_Command7_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_Command7_Click
```

```
End Sub
```

```
Private Sub Command8_Click()
```

```
On Error GoTo Err_Command8_Click
```

```
DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
```

```
Exit_Command8_Click:
```

```
Exit Sub
```

```
Err_Command8_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_Command8_Click
```

```
End Sub
```

```
Private Sub Command9_Click()  
On Error GoTo Err_Command9_Click  
    DoCmd.Close  
Exit_Command9_Click:  
    Exit Sub  
Err_Command9_Click:  
    MsgBox Err.Description  
    Resume Exit_Command9_Click  
End Sub
```

```
Private Sub Form_Open(Cancel As Integer)  
Dim stDocName As String  
Dim LinkCriteria As String  
stDocName = "EditDoctors"  
DoCmd.OpenForm stDocName, , , LinkCriteria, A_ADD  
Exit_Command79_Click:  
    Exit Sub  
Err_Command79_Click:  
    MsgBox Err.Description  
    Resume Exit_Command79_Click  
End Sub
```

```
Private Sub Command22_Click()  
On Error GoTo Err_Command22_Click  
    Dim stDocName As String  
    stDocName = "DetailDoc"  
    DoCmd.OpenReport stDocName, acNormal, , "[iddoctor]=  
forms![AddDoctors].[iddoctor]"  
Exit_Command22_Click:
```

```
Exit Sub
Err_Command22_Click:
    MsgBox Err.Description
    Resume Exit_Command22_Click
End Sub
```

3.2 คลาสโมดูลของ Form CheckOut

```
Private Sub Command2_Click()
On Error GoTo Err_Command2_Click
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
Exit_Command2_Click:
    Exit Sub
Err_Command2_Click:
    MsgBox Err.Description
    Resume Exit_Command2_Click
End Sub
```

3.3 คลาสโมดูลของ Form Comein1

```
Private Sub Command10_Click()
On Error GoTo Err_Command10_Click
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
Exit_Command10_Click:
    Exit Sub
```

```
Err_Command10_Click:
    MsgBox Err.Description
    Resume Exit_Command10_Click
End Sub

Private Sub Command12_Click()
On Error GoTo Err_Command12_Click
    DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
    DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
Exit_Command12_Click:
    Exit Sub
Err_Command12_Click:
    MsgBox Err.Description
    Resume Exit_Command12_Click
End Sub

Private Sub Command15_Click()
On Error GoTo Err_Command15_Click
    DoCmd.Close
Exit_Command15_Click:
    Exit Sub
Err_Command15_Click:
    MsgBox Err.Description
    Resume Exit_Command15_Click
End Sub

Private Sub Form_Open(Cancel As Integer)
Dim stDocName As String
stDocName = "Comein1"
DoCmd.OpenForm stDocName, acNormal, , "[code] = [กรอกเลขประจำตัวผู้ปวย]"
If IsNull(code) = 0 Then
```



```

Else
    MsgBox "ไม่มีข้อมูลในระบบ", 0, "แจ้งให้ทราบ"
    DoCmd.Close
End If
End Sub

```

3.4 คลาสโมดูลของ Form Comein2

```

Private Sub Command37_Click()
On Error GoTo Err_Command37_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "PickCome"
    stLinkCriteria = "[codecomein]=" & "" & Me![codecomein] & ""
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command37_Click:
    Exit Sub
Err_Command37_Click:
    MsgBox Err.Description
    Resume Exit_Command37_Click
End Sub

```

```

Private Sub Command38_Click()
On Error GoTo Err_Command38_Click
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
Exit_Command38_Click:
    Exit Sub
Err_Command38_Click:
    MsgBox Err.Description
    Resume Exit_Command38_Click

```

End Sub

Private Sub Command39_Click()

On Error GoTo Err_Command39_Click

If MsgBox("คุณแน่ใจว่าต้องการลบข้อมูลการเข้าพักนี้", 36, "ยืนยัน") = 6 Then

 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70

 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70

End If

Exit_Command39_Click:

 Exit Sub

Err_Command39_Click:

 MsgBox Err.Description

 Resume Exit_Command39_Click

End Sub

3.5 กลาสโมดูลของ Form Comein3

Private Sub Command3_Click()

On Error GoTo Err_Command3_Click

 DoCmd.Close

Exit_Command3_Click:

 Exit Sub

Err_Command3_Click:

 MsgBox Err.Description

 Resume Exit_Command3_Click

End Sub

3.6 กลาสโมดูลของ Form CopyComein2

Private Sub Command16_Click()

On Error GoTo Err_Command16_Click

```
DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
Exit_Command16_Click:
    Exit Sub
Err_Command16_Click:
    MsgBox Err.Description
    Resume Exit_Command16_Click
End Sub

Private Sub Command21_Click()
On Error GoTo Err_Command21_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "DetailDoctor"
    stLinkCriteria = "[iddoctor]=" & "" & Me![Combo19] & ""
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command21_Click:
    Exit Sub

Err_Command21_Click:
    MsgBox Err.Description
    Resume Exit_Command21_Click
End Sub

Sub Combo22_AfterUpdate()
    ' Find the record that matches the control.
    Me.RecordsetClone.FindFirst "[code] = " & Me![Combo22] & ""
    Me.Bookmark = Me.RecordsetClone.Bookmark
End Sub

Private Sub Command25_Click()
```

```
On Error GoTo Err_Command25_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Comein3"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
```

```
Exit_Command25_Click:
```

```
    Exit Sub
```

```
Err_Command25_Click:
```

```
    MsgBox Err.Description
```

```
    Resume Exit_Command25_Click
```

```
End Sub
```

```
Private Sub Command27_Click()
```

```
On Error GoTo Err_Command27_Click
```

```
    Dim stDocName As String
```

```
    Dim stLinkCriteria As String
```

```
    stDocName = "Comein3"
```

```
    stLinkCriteria = "[codecomein]=" & "" & Me!["codecomein"] & ""
```

```
    DoCmd.OpenForm stDocName, , , stLinkCriteria
```

```
Exit_Command27_Click:
```

```
    Exit Sub
```

```
Err_Command27_Click:
```

```
    MsgBox Err.Description
```

```
    Resume Exit_Command27_Click
```

```
End Sub
```

```
Private Sub Command32_Click()
```

```
On Error GoTo Err_Command32_Click
```

```
    Dim stDocName As String
```

```
    Dim stLinkCriteria As String
```

```
    stDocName = "DetailDoctor"
```

```

stLinkCriteria = "[iddoctor]=" & "" & Me![Combo28] & ""
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command32_Click:
    Exit Sub
Err_Command32_Click:
    MsgBox Err.Description
    Resume Exit_Command32_Click
End Sub

```

3.7 คลาสโมดูลของ Form DetailDoctor

```

Private Sub Command11_Click()
On Error GoTo Err_Command11_Click
    DoCmd.Close
Exit_Command11_Click:
    Exit Sub
Err_Command11_Click:
    MsgBox Err.Description
    Resume Exit_Command11_Click
End Sub

```

3.8 คลาสโมดูลของ Form Doctor2

```

Private Sub Command3_Click()
On Error GoTo Err_Command3_Click
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
Exit_Command3_Click:
    Exit Sub
Err_Command3_Click:
    MsgBox Err.Description
    Resume Exit_Command3_Click

```

End Sub

```
Private Sub Command4_Click()  
On Error GoTo Err_Command4_Click  
    DoCmd.Close  
Exit_Command4_Click:  
    Exit Sub  
Err_Command4_Click:  
    MsgBox Err.Description  
    Resume Exit_Command4_Click  
End Sub
```

```
Private Sub Command14_Click()  
On Error GoTo Err_Command14_Click  
    Dim stDocName As String  
    Dim stLinkCriteria As String  
    stDocName = "Doctors"  
    stLinkCriteria = "[iddoctor]=" & "" & Me![iddoctor] & ""  
    DoCmd.OpenForm stDocName, , , stLinkCriteria  
Exit_Command14_Click:  
    Exit Sub  
Err_Command14_Click:  
    MsgBox Err.Description  
    Resume Exit_Command14_Click  
End Sub
```

```
Private Sub Form_Open(Cancel As Integer)  
    Dim stDocName As String  
    stDocName = "Doctor2"  
    DoCmd.OpenForm stDocName, acNormal, , "[iddoctor] = [กรอกรหัสแพทย์]"
```

End Sub

3.9 คลาสโมดูลของ Form Doctors

```
Private Sub Command7_Click()  
On Error GoTo Err_Command7_Click  
    DoCmd.GoToRecord , , acNewRec  
Exit_Command7_Click:  
    Exit Sub  
Err_Command7_Click:  
    MsgBox Err.Description  
    Resume Exit_Command7_Click  
End Sub
```

```
Private Sub Command8_Click()  
On Error GoTo Err_Command8_Click  
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70  
Exit_Command8_Click:  
    Exit Sub  
Err_Command8_Click:  
    MsgBox Err.Description  
    Resume Exit_Command8_Click  
End Sub
```

```
Private Sub Command9_Click()  
On Error GoTo Err_Command9_Click  
    DoCmd.Close  
Exit_Command9_Click:  
    Exit Sub  
  
Err_Command9_Click:
```

```

    MsgBox Err.Description
    Resume Exit_Command9_Click
End Sub

Private Sub Form_Open(Cancel As Integer)
Dim stDocName As String
stDocName = "Doctors"
DoCmd.OpenForm stDocName, acNormal, , "[iddoctor] = [กรอกรหัสแพทย์]"
End Sub

Private Sub Command11_Click()
On Error GoTo Err_Command11_Click
    DoCmd.Close
Exit_Command11_Click:
    Exit Sub
Err_Command11_Click:
    MsgBox Err.Description
    Resume Exit_Command11_Click
End Sub

```

3.10 คลาสโมดูลของ Form EditDoctors

```

Private Sub Command7_Click()
On Error GoTo Err_Command7_Click
    DoCmd.GoToRecord , , acNewRec
Exit_Command7_Click:
    Exit Sub
Err_Command7_Click:
    MsgBox Err.Description
    Resume Exit_Command7_Click

```


End Sub

Private Sub Command8_Click()

On Error GoTo Err_Command8_Click

DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70

Exit_Command8_Click:

Exit Sub

Err_Command8_Click:

MsgBox Err.Description

Resume Exit_Command8_Click

End Sub

Private Sub Command9_Click()

On Error GoTo Err_Command9_Click

DoCmd.Close

Exit_Command9_Click:

Exit Sub

Err_Command9_Click:

MsgBox Err.Description

Resume Exit_Command9_Click

End Sub

Private Sub Form_Open(Cancel As Integer)

Dim stDocName As String

Dim LinkCriteria As String

stDocName = "EditDoctors"

DoCmd.OpenForm stDocName, acNormal, , "[iddoctor]= [กรอกรหัสแพทย์ที่ต้องการแก้ไข]"

End Sub

Private Sub คำสั่ง29_Click()

On Error GoTo Err_คำสั่ง29_Click

```

If MsgBox("คุณแน่ใจว่าต้องการลบเรคคอร์ดนี้", 36, "ยืนยัน") = 6 Then
    DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
    DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
End If
Exit_คำสั่ง29_Click:
    Exit Sub

Err_คำสั่ง29_Click:
    MsgBox Err.Description
    Resume Exit_คำสั่ง29_Click
End Sub

Private Sub คำสั่ง30_Click()
On Error GoTo Err_คำสั่ง30_Click
    Dim stDocName As String
    stDocName = "DetailDoc"
    DoCmd.OpenReport stDocName, acNormal, , "[iddoctor]=
forms![EditDoctors].[iddoctor]"
Exit_คำสั่ง30_Click:
    Exit Sub
Err_คำสั่ง30_Click:
    MsgBox Err.Description
    Resume Exit_คำสั่ง30_Click
End Sub

```

3.11 คลาสโมดูลของ Form EditPatient

```

Private Sub ค้นหา_Click()
On Error GoTo Err_ค้นหา_Click
    Screen.PreviousControl.SetFocus
    DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70
Exit_ค้นหา_Click:

```

```
Exit Sub
Err_ค้นหา_Click:
    MsgBox Err.Description
    Resume Exit_ค้นหา_Click
End Sub

Private Sub Command108_Click()
On Error GoTo Err_Command108_Click
If MsgBox("คุณแน่ใจว่าต้องการลบเรคคอร์ดนี้", 36, "ยืนยัน") = 6 Then

    DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
    DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
End If
Exit_Command108_Click:
    Exit Sub
Err_Command108_Click:
    MsgBox Err.Description
    Resume Exit_Command108_Click
End Sub

Private Sub Command109_Click()
On Error GoTo Err_Command109_Click
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
Exit_Command109_Click:
    Exit Sub
Err_Command109_Click:
    MsgBox Err.Description
    Resume Exit_Command109_Click
End Sub

Private Sub Command110_Click()
```

```
On Error GoTo Err_Command110_Click
```

```
DoCmd.Close
```

```
Exit_Command110_Click:
```

```
Exit Sub
```

```
Err_Command110_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_Command110_Click
```

```
End Sub
```

```
Private Sub Form_Open(Cancel As Integer)
```

```
Dim stDocName As String
```

```
stDocName = "EditPatient"
```

```
DoCmd.OpenForm stDocName, acNormal, , "[code] = [กรอกเลขประจำตัวผู้ป่วย]"
```

```
If IsNull(code) = 0 Then
```

```
Else
```

```
MsgBox "ไม่มีข้อมูลในระบบ", 0, "แจ้งให้ทราบ"
```

```
DoCmd.Close
```

```
End If
```

```
End Sub
```

3.12 คลาสโมดูลของ Form Main

```
Private Sub Command0_Click()
```

```
On Error GoTo Err_Command0_Click
```

```
Dim stDocName As String
```

```
Dim stLinkCriteria As String
stDocName = "SearchComein"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command0_Click:
Exit Sub
Err_Command0_Click:
MsgBox Err.Description
Resume Exit_Command0_Click
End Sub
```

```
Private Sub Command11_Click()
On Error GoTo Err_Command2_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Patient"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command2_Click:
Exit Sub
Err_Command2_Click:
MsgBox Err.Description
Resume Exit_Command2_Click
End Sub
```

```
Private Sub Command1_Click()
On Error GoTo Err_Command1_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Patientnew"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command1_Click:
Exit Sub
```

```
Err_Command1_Click:
    MsgBox Err.Description
    Resume Exit_Command1_Click
End Sub

Private Sub Command2_Click()
On Error GoTo Err_Command2_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "EditPatient"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command2_Click:
    Exit Sub
Err_Command2_Click:
    MsgBox Err.Description
    Resume Exit_Command2_Click
End Sub

Private Sub Command3_Click()
Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Comein1"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command2_Click:
    Exit Sub
Err_Command2_Click:
    MsgBox Err.Description
    Resume Exit_Command2_Click
End Sub

Private Sub Command5_Click()
```

```
comDialog.DialogTitle = "ตัวอย่างการเลือกเครื่องพิมพ์"  
comDialog.Copies = 3  
comDialog.ShowPrinter  
End Sub
```

```
Private Sub Command8_Click()  
On Error GoTo Err_Command8_Click  
    Dim stDocName As String  
    Dim stLinkCriteria As String  
    stDocName = "Patient(name)"  
    DoCmd.OpenForm stDocName, , , stLinkCriteria  
Exit_Command8_Click:  
    Exit Sub  
Err_Command8_Click:  
    MsgBox Err.Description  
    Resume Exit_Command8_Click  
End Sub
```

```
Private Sub Form_Current()  
DoCmd.Maximize  
DoCmd.GoToControl "input_code"  
End Sub
```

```
Private Sub Form_Open(Cancel As Integer)  
DoCmd.Maximize  
End Sub
```

```
Private Sub input_code_AfterUpdate()  
If Not IsNull(Me![input_code]) Then  
    Let code_Fail = Me![code]  
    DoCmd.GoToControl "code"
```

```

DoCmd.FindRecord Me![input_code]
If Me![code] = code_Fail Then
    MsgBox "ไม่มีข้อมูลนักศึกษา" & Chr(10) & Chr(13) & "กรุณาใส่รหัสนักศึกษาใหม่", 48,
"ไม่มีข้อมูล"
    Let Me![input_code] = ""
    DoCmd.GoToControl "input_code"
Else
Let set_find = -1
DoCmd.Close A_FORM, "Menu"
DoCmd.OpenForm "Patient"
End If
End If
End Sub

```

```

Private Sub Command12_Click()
On Error GoTo Err_Command12_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Patientnew"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command12_Click:
    Exit Sub
Err_Command12_Click:
    MsgBox Err.Description
    Resume Exit_Command12_Click
End Sub

```

```

Private Sub Command13_Click()
On Error GoTo Err_Command13_Click
    Dim stDocName As String
    Dim stLinkCriteria As String

```



```
stDocName = "EditPatient"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command13_Click:
Exit Sub

Err_Command13_Click:
MsgBox Err.Description
Resume Exit_Command13_Click
End Sub

Private Sub Command15_Click()
On Error GoTo Err_Command15_Click
If MsgBox("คุณแน่ใจหรือว่าจะออกจากโปรแกรม", 36, "ยืนยัน") = 6 Then
DoCmd.Quit
Else
End If
Exit_Command15_Click:
Exit Sub
Err_Command15_Click:
MsgBox Err.Description
Resume Exit_Command15_Click
End Sub

Private Sub Command45_Click()
On Error GoTo Err_Command45_Click
Dim stDocName As String
stDocName = "Patientnew"
Screen.PreviousControl.SetFocus
DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70
Exit_Command45_Click:
Exit Sub
```

```
Err_Command45_Click:
    MsgBox Err.Description
    Resume Exit_Command45_Click
End Sub

Private Sub Command46_Click()
On Error GoTo Err_Command46_Click

    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "SearchComein(name)"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command46_Click:
    Exit Sub
Err_Command46_Click:
    MsgBox Err.Description
    Resume Exit_Command46_Click
End Sub

Private Sub Command47_Click()
On Error GoTo Err_Command47_Click

    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "SeaCome(No)"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command47_Click:
    Exit Sub
Err_Command47_Click:
    MsgBox Err.Description
    Resume Exit_Command47_Click
End Sub
```

3.13 กลาสโมดูลของ Form Patient

```
Private Sub Command62_Click()
On Error GoTo Err_Command62_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Qdetail"
    stLinkCriteria = "[code]=" & "" & Me![code] & ""
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command62_Click:
    Exit Sub
```

```
Err_Command62_Click:
    MsgBox Err.Description
    Resume Exit_Command62_Click
End Sub
```

```
Private Sub Popup1_Click()
On Error GoTo Err_Popup1_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Qdetail"
    stLinkCriteria = "[code]=" & "" & Me![code] & ""
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Popup1_Click:
    Exit Sub
Err_Popup1_Click:
    MsgBox Err.Description
    Resume Exit_Popup1_Click
End Sub
```

```
Private Sub Command66_Click()  
On Error GoTo Err_Command66_Click  
    Dim stDocName As String  
    Dim stLinkCriteria As String  
    stDocName = "Patientnew"  
    DoCmd.OpenForm stDocName, , , stLinkCriteria  
Exit_Command66_Click:  
    Exit Sub  
Err_Command66_Click:  
    MsgBox Err.Description  
    Resume Exit_Command66_Click  
End Sub
```

```
Private Sub Command68_Click()  
On Error GoTo Err_Command68_Click  
    Dim stDocName As String  
    Dim stLinkCriteria As String  
    stDocName = "EditPatient"  
    DoCmd.OpenForm stDocName, , , stLinkCriteria  
Exit_Command68_Click:  
    Exit Sub  
Err_Command68_Click:  
    MsgBox Err.Description  
    Resume Exit_Command68_Click  
End Sub
```

```
Private Sub Command69_Click()  
On Error GoTo Err_Command69_Click  
    Dim stDocName As String  
    Dim stLinkCriteria As String
```

```
stDocName = "EditPatient"
stLinkCriteria = "[code]=" & "" & Me![code] & ""
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command69_Click:
Exit Sub
Err_Command69_Click:
MsgBox Err.Description
Resume Exit_Command69_Click
End Sub

Private Sub Command71_Click()
On Error GoTo Err_Command71_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Patientnew"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command71_Click:
Exit Sub
Err_Command71_Click:
MsgBox Err.Description
Resume Exit_Command71_Click
End Sub

Private Sub Command72_Click()
On Error GoTo Err_Command72_Click
DoCmd.Close
Exit_Command72_Click:
Exit Sub
Err_Command72_Click:
MsgBox Err.Description
```

```
Resume Exit_Command72_Click
End Sub

Private Sub Command73_Click()
On Error GoTo Err_Command73_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Patientnew"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command73_Click:
    Exit Sub
Err_Command73_Click:
    MsgBox Err.Description
    Resume Exit_Command73_Click
End Sub

Private Sub Command74_Click()
On Error GoTo Err_Command74_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "EditPatient"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command74_Click:
    Exit Sub
Err_Command74_Click:
    MsgBox Err.Description
    Resume Exit_Command74_Click
End Sub

Private Sub Command75_Click()
```

```
On Error GoTo Err_Command75_Click
    Dim stDocName As String
    stDocName = "AboutPatient"
    DoCmd.OpenReport stDocName, acPreview, , "[code] = forms![Aboutpatient].[code]"
Exit_Command75_Click:
    Exit Sub
Err_Command75_Click:
    MsgBox Err.Description
    Resume Exit_Command75_Click
End Sub
```

```
Private Sub Command76_Click()
On Error GoTo Err_Command76_Click
    Dim stDocName As String
    stDocName = "AboutPatient"
    DoCmd.OpenReport stDocName, acNormal, , "[code] =forms![Patient].[code]"
Exit_Command76_Click:
    Exit Sub
Err_Command76_Click:
    MsgBox Err.Description
    Resume Exit_Command76_Click
End Sub
```

```
Private Sub Command77_Click()
On Error GoTo Err_Command77_Click
    Dim stDocName As String

    stDocName = "Cards"
    DoCmd.OpenReport stDocName, acNormal, , "[code]= forms![Patient].[code]"
Exit_Command77_Click:
    Exit Sub
```

```
Err_Command77_Click:
```

```
    MsgBox Err.Description
```

```
    Resume Exit_Command77_Click
```

```
End Sub
```

```
Private Sub Command79_Click()
```

```
On Error GoTo Err_Command79_Click
```

```
    Dim stDocName As String
```

```
    stDocName = "Patient"
```

```
    DoCmd.OpenForm stDocName, acNormal, , "[code] = [กรอกเลขประจำตัวผู้ป่วย]"
```

```
    If MsgBox("ไม่มีข้อมูลในระบบโปรดกรอกข้อมูลให้ถูกต้อง", 36, "ยืนยัน") = 6 Then
```

```
        DoCmd.Close
```

```
Else
```

```
End If
```

```
Exit_Command79_Click:
```

```
    Exit Sub
```

```
Err_Command79_Click:
```

```
    MsgBox Err.Description
```

```
    Resume Exit_Command79_Click
```

```
End Sub
```

```
Private Sub Command80_Click()
```

```
On Error GoTo Err_Command80_Click
```

```
    Dim stDocName As String
```

```
    stDocName = "Patient(name)"
```

```
    DoCmd.OpenForm stDocName, acNormal
```

```
Exit_Command80_Click:
```

```
    Exit Sub
```

```
Err_Command80_Click:
```

```
    MsgBox Err.Description
```

```
    Resume Exit_Command80_Click
```


End Sub

```
Private Sub Form_Open(Cancel As Integer)
```

```
Dim stDocName As String
```

```
stDocName = "Patient"
```

```
DoCmd.OpenForm stDocName, acNormal, , "[code] = [กรอกเลขประจำตัวผู้ป่วย]"
```

```
If IsNull(code) = 0 Then
```

```
Else
```

```
MsgBox "ไม่มีข้อมูลในระบบ", 0, "แจ้งให้ทราบ"
```

```
DoCmd.Close
```

```
End If
```

```
End Sub
```

```
Private Sub Command81_Click()
```

```
On Error GoTo Err_Command81_Click
```

```
Dim stDocName As String
```

```
stDocName = "AboutPatient"
```

```
DoCmd.OpenReport stDocName, acPreview, , "[code]= forms![Patient].[code]"
```

```
Exit_Command81_Click:
```

```
Exit Sub
```

```
Err_Command81_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_Command81_Click
```

```
End Sub
```

```
Private Sub Command82_Click()
```

```
On Error GoTo Err_Command82_Click
```

```
Dim stDocName As String
```

```
Dim stLinkCriteria As String
```

```
stDocName = "Qcard"
```

```
stLinkCriteria = "[code]=" & "" & Me![code] & ""
```

```

DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command82_Click:
Exit Sub

Err_Command82_Click:
MsgBox Err.Description
Resume Exit_Command82_Click
End Sub

```

3.14 คลาสโมดูลของ Form Patient(name)

```

Private Sub Command62_Click()
On Error GoTo Err_Command62_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Qdetail"
stLinkCriteria = "[code]=" & "" & Me![code] & ""
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command62_Click:
Exit Sub
Err_Command62_Click:
MsgBox Err.Description
Resume Exit_Command62_Click
End Sub

```

```

Private Sub Popup1_Click()
On Error GoTo Err_Popup1_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Qdetail"
stLinkCriteria = "[code]=" & "" & Me![code] & ""
DoCmd.OpenForm stDocName, , , stLinkCriteria

```

Exit_Popup1_Click:

Exit Sub

Err_Popup1_Click:

MsgBox Err.Description

Resume Exit_Popup1_Click

End Sub

Private Sub Command66_Click()

On Error GoTo Err_Command66_Click

Dim stDocName As String

Dim stLinkCriteria As String

stDocName = "Patientnew"

DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command66_Click:

Exit Sub

Err_Command66_Click:

MsgBox Err.Description

Resume Exit_Command66_Click

End Sub

Private Sub Command68_Click()

On Error GoTo Err_Command68_Click

Dim stDocName As String

Dim stLinkCriteria As String

stDocName = "EditPatient"

DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command68_Click:

Exit Sub

Err_Command68_Click:

MsgBox Err.Description

```
Resume Exit_Command68_Click
End Sub

Private Sub Command69_Click()
On Error GoTo Err_Command69_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "EditPatient"
stLinkCriteria = "[code]=" & "" & Me![code] & ""
DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command69_Click:
Exit Sub
Err_Command69_Click:
MsgBox Err.Description
Resume Exit_Command69_Click
End Sub

Private Sub Command71_Click()
On Error GoTo Err_Command71_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Patientnew"
DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command71_Click:
Exit Sub
Err_Command71_Click:
MsgBox Err.Description
Resume Exit_Command71_Click
End Sub
```

```
Private Sub Command72_Click()  
On Error GoTo Err_Command72_Click  
    DoCmd.Close  
Exit_Command72_Click:  
    Exit Sub  
Err_Command72_Click:  
    MsgBox Err.Description  
    Resume Exit_Command72_Click  
End Sub
```

```
Private Sub Command73_Click()  
On Error GoTo Err_Command73_Click  
    Dim stDocName As String  
    Dim stLinkCriteria As String  
  
    stDocName = "Patientnew"  
    DoCmd.OpenForm stDocName, , , stLinkCriteria  
Exit_Command73_Click:  
    Exit Sub  
Err_Command73_Click:  
    MsgBox Err.Description  
    Resume Exit_Command73_Click  
End Sub
```

```
Private Sub Command74_Click()  
On Error GoTo Err_Command74_Click  
    Dim stDocName As String  
    Dim stLinkCriteria As String  
    stDocName = "EditPatient"  
    DoCmd.OpenForm stDocName, , , stLinkCriteria  
Exit_Command74_Click:
```

```
Exit Sub
Err_Command74_Click:
    MsgBox Err.Description
    Resume Exit_Command74_Click
End Sub

Private Sub Command75_Click()
On Error GoTo Err_Command75_Click
    Dim stDocName As String
    stDocName = "AboutPatient"
    DoCmd.OpenReport stDocName, acPreview, , "[code] = forms![Aboutpatient].[code]"
Exit_Command75_Click:
    Exit Sub
Err_Command75_Click:
    MsgBox Err.Description
    Resume Exit_Command75_Click
End Sub

Private Sub Command76_Click()
On Error GoTo Err_Command76_Click
    Dim stDocName As String
    stDocName = "AboutPatient"
    DoCmd.OpenReport stDocName, acNormal, , "[code] =forms![Patient(name)].[code]"
Exit_Command76_Click:
    Exit Sub
Err_Command76_Click:
    MsgBox Err.Description
    Resume Exit_Command76_Click
End Sub

Private Sub Command77_Click()
```

```
On Error GoTo Err_Command77_Click
    Dim stDocName As String
    stDocName = "Cards"
    DoCmd.OpenReport stDocName, acNormal, , "[code]= forms![Patient(Name)].[code]"
```

```
Exit_Command77_Click:
```

```
    Exit Sub
```

```
Err_Command77_Click:
```

```
    MsgBox Err.Description
```

```
    Resume Exit_Command77_Click
```

```
End Sub
```

```
Private Sub Command79_Click()
```

```
On Error GoTo Err_Command79_Click
```

```
    Dim stDocName As String
```

```
    stDocName = "Patient"
```

```
    DoCmd.OpenForm stDocName, acNormal
```

```
Exit_Command79_Click:
```

```
    Exit Sub
```

```
Err_Command79_Click:
```

```
    MsgBox Err.Description
```

```
    Resume Exit_Command79_Click
```

```
End Sub
```

```
Private Sub Command80_Click()
```

```
On Error GoTo Err_Command80_Click
```

```
    Dim stDocName As String
```

```
    stDocName = "Patient(name)"
```

```
    DoCmd.OpenForm stDocName, acNormal
```

```
Exit_Command80_Click:
```

```
    Exit Sub
```

```
Err_Command80_Click:
```

```

    MsgBox Err.Description
    Resume Exit_Command80_Click
End Sub

Private Sub Form_Open(Cancel As Integer)
    Dim stDocName As String
    stDocName = "Patient(name)"
    DoCmd.OpenForm stDocName, acNormal, , "[name] = [พิมพ์ชื่อ-สกุลผู้ป่วยที่ต้องการหา]"
End Sub

Private Sub Command81_Click()
    On Error GoTo Err_Command81_Click
    Dim stDocName As String
    stDocName = "AboutPatient"
    DoCmd.OpenReport stDocName, acPreview, , "[code]= forms![Patient(name)].[code]"
Exit_Command81_Click:
    Exit Sub
Err_Command81_Click:
    MsgBox Err.Description
    Resume Exit_Command81_Click
End Sub

Private Sub Command82_Click()
    On Error GoTo Err_Command82_Click
    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "Qcard"
    stLinkCriteria = "[code]=" & "" & Me![code] & ""
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command82_Click:

```



```

Exit Sub
Err_Command82_Click:
    MsgBox Err.Description
    Resume Exit_Command82_Click
End Sub

Private Sub Command276_Click()
On Error GoTo Err_Command276_Click

    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Patient(name)"
    DoCmd.OpenForm stDocName, acNormal, , "[name]= [กรอกชื่อ-สกุลผู้ป่วย]"
Exit_Command276_Click:
    Exit Sub
Err_Command276_Click:
    MsgBox Err.Description
    Resume Exit_Command276_Click
End Sub

```

3.15 คลาสโมดูลของ Form Patient (No)

```

Private Sub Command62_Click()
On Error GoTo Err_Command62_Click

    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Qdetail"
    stLinkCriteria = "[code]=" & "" & Me![code] & ""
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command62_Click:
    Exit Sub

```

```
Err_Command62_Click:
    MsgBox Err.Description
    Resume Exit_Command62_Click
End Sub

Private Sub Popup1_Click()
On Error GoTo Err_Popup1_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Qdetail"
    stLinkCriteria = "[code]=" & "" & Me![code] & ""
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Popup1_Click:
    Exit Sub
Err_Popup1_Click:
    MsgBox Err.Description
    Resume Exit_Popup1_Click
End Sub

Private Sub Command66_Click()
On Error GoTo Err_Command66_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Patientnew"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command66_Click:
    Exit Sub
Err_Command66_Click:
    MsgBox Err.Description
    Resume Exit_Command66_Click
End Sub
```

```
Private Sub Command68_Click()  
On Error GoTo Err_Command68_Click  
  
    Dim stDocName As String  
    Dim stLinkCriteria As String  
    stDocName = "EditPatient"  
    DoCmd.OpenForm stDocName, , , stLinkCriteria  
Exit_Command68_Click:  
    Exit Sub  
Err_Command68_Click:  
    MsgBox Err.Description  
    Resume Exit_Command68_Click  
End Sub
```

```
Private Sub Command69_Click()  
On Error GoTo Err_Command69_Click  
  
    Dim stDocName As String  
    Dim stLinkCriteria As String  
    stDocName = "EditPatient"  
    stLinkCriteria = "[code]=" & "" & Me![code] & ""  
    DoCmd.OpenForm stDocName, , , stLinkCriteria  
Exit_Command69_Click:  
    Exit Sub  
Err_Command69_Click:  
    MsgBox Err.Description  
    Resume Exit_Command69_Click  
End Sub
```

```
Private Sub Command71_Click()  
On Error GoTo Err_Command71_Click
```

```
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Patientnew"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command71_Click:
Exit Sub
Err_Command71_Click:
MsgBox Err.Description
Resume Exit_Command71_Click
End Sub

Private Sub Command72_Click()
On Error GoTo Err_Command72_Click
DoCmd.Close
Exit_Command72_Click:
Exit Sub
Err_Command72_Click:
MsgBox Err.Description
Resume Exit_Command72_Click
End Sub

Private Sub Command73_Click()
On Error GoTo Err_Command73_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Patientnew"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command73_Click:
Exit Sub
Err_Command73_Click:
MsgBox Err.Description
```

```
Resume Exit_Command73_Click
End Sub

Private Sub Command74_Click()
On Error GoTo Err_Command74_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "EditPatient"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command74_Click:
    Exit Sub

Err_Command74_Click:
    MsgBox Err.Description
    Resume Exit_Command74_Click
End Sub

Private Sub Command75_Click()
On Error GoTo Err_Command75_Click
    Dim stDocName As String
    stDocName = "AboutPatient"
    DoCmd.OpenReport stDocName, acPreview, , "[code] = forms![Aboutpatient].[code]"
Exit_Command75_Click:
    Exit Sub

Err_Command75_Click:
    MsgBox Err.Description
    Resume Exit_Command75_Click
End Sub

Private Sub Command76_Click()
On Error GoTo Err_Command76_Click
```

```
Dim stDocName As String
stDocName = "AboutPatient"
DoCmd.OpenReport stDocName, acNormal, , "[code] =forms![Patient(No)].[code]"
Exit_Command76_Click:
Exit Sub
Err_Command76_Click:
MsgBox Err.Description
Resume Exit_Command76_Click
End Sub

Private Sub Command77_Click()
On Error GoTo Err_Command77_Click
Dim stDocName As String
stDocName = "Cards"
DoCmd.OpenReport stDocName, acNormal, , "[code]= forms![Patient(No)].[code]"

Exit_Command77_Click:
Exit Sub
Err_Command77_Click:
MsgBox Err.Description
Resume Exit_Command77_Click
End Sub

Private Sub Command79_Click()
On Error GoTo Err_Command79_Click
Dim stDocName As String
stDocName = "Patient"
DoCmd.OpenForm stDocName, acNormal
Exit_Command79_Click:
Exit Sub
Err_Command79_Click:
```

```

    MsgBox Err.Description
    Resume Exit_Command79_Click
End Sub

Private Sub Command81_Click()
On Error GoTo Err_Command81_Click
    Dim stDocName As String
    stDocName = "AboutPatient"
    DoCmd.OpenReport stDocName, acPreview, , "[code]= forms![Patient(No)].[code]"
Exit_Command81_Click:
    Exit Sub
Err_Command81_Click:
    MsgBox Err.Description
    Resume Exit_Command81_Click
End Sub

Private Sub Command82_Click()
On Error GoTo Err_Command82_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Qcard"
    stLinkCriteria = "[code]=" & "" & Me![code] & ""
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command82_Click:
    Exit Sub
Err_Command82_Click:
    MsgBox Err.Description
    Resume Exit_Command82_Click
End Sub

Private Sub คำสั่ง289_Click()

```

```

On Error GoTo Err_คำสั่ง289_Click
    Screen.PreviousControl.SetFocus
    DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70
Exit_คำสั่ง289_Click:
    Exit Sub
Err_คำสั่ง289_Click:
    MsgBox Err.Description
    Resume Exit_คำสั่ง289_Click
End Sub

```

คลาสโมดูลของ Form Patientnew

```

Private Sub ค้นหา_Click()
On Error GoTo Err_ค้นหา_Click
    Screen.PreviousControl.SetFocus
    DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70
Exit_ค้นหา_Click:
    Exit Sub
Err_ค้นหา_Click:
    MsgBox Err.Description
    Resume Exit_ค้นหา_Click
End Sub

```

```

Private Sub Command108_Click()
On Error GoTo Err_Command108_Click
If MsgBox("คุณแน่ใจว่าต้องการลบเรคคอร์ดนี้", 36, "ยืนยัน") = 6 Then
    DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
    DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
Else
End If
Exit_Command108_Click:

```



```
Exit Sub
Err_Command108_Click:
    MsgBox Err.Description
    Resume Exit_Command108_Click
End Sub

Private Sub Command109_Click()
On Error GoTo Err_Command109_Click
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
Exit_Command109_Click:
    Exit Sub
Err_Command109_Click:
    MsgBox Err.Description
    Resume Exit_Command109_Click
End Sub

Private Sub Command110_Click()
On Error GoTo Err_Command110_Click
    DoCmd.Close
Exit_Command110_Click:
    Exit Sub
Err_Command110_Click:
    MsgBox Err.Description
    Resume Exit_Command110_Click
End Sub

Private Sub Command111_Click()
On Error GoTo Err_Command111_Click
    DoCmd.GoToRecord , , acNewRec
Exit_Command111_Click:
    Exit Sub
```

```

Err_Command111_Click:
    MsgBox Err.Description
    Resume Exit_Command111_Click
End Sub

```

3.16 คลาสโมดูลของ Form PickComein

```

Private Sub Command5_Click()
On Error GoTo Err_Command5_Click
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
Exit_Command5_Click:
    Exit Sub
Err_Command5_Click:
    MsgBox Err.Description
    Resume Exit_Command5_Click
End Sub

```

```

Private Sub Command6_Click()
On Error GoTo Err_Command6_Click
    DoCmd.Close
Exit_Command6_Click:
    Exit Sub
Err_Command6_Click:
    MsgBox Err.Description
    Resume Exit_Command6_Click
End Sub

```

3.17 คลาสโมดูลของ Form Qcard

```

Private Sub Command13_Click()
On Error GoTo Err_Command13_Click

```

```

Dim stDocName As String
stDocName = "Qcard"
DoCmd.OpenReport stDocName, acNormal
Exit_Command13_Click:
Exit Sub
Err_Command13_Click:
MsgBox Err.Description
Resume Exit_Command13_Click
End Sub

Private Sub Command15_Click()
On Error GoTo Err_Command15_Click
Dim stDocName As String
stDocName = "Qcard "
DoCmd.OpenReport stDocName, acNormal, , "[code] =forms![Qcard].[code]"
Exit_Command15_Click:
Exit Sub
Err_Command15_Click:
MsgBox Err.Description
Resume Exit_Command15_Click
End Sub

```

3.18 คลาสโมดูลของ Form Qcard2

```

Private Sub Command4_Click()
On Error GoTo Err_Command4_Click
DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
Exit_Command4_Click:

```

```
Exit Sub
Err_Command4_Click:
    MsgBox Err.Description
    Resume Exit_Command4_Click
End Sub

Private Sub Command5_Click()
On Error GoTo Err_Command5_Click
    DoCmd.Close
Exit_Command5_Click:
    Exit Sub
Err_Command5_Click:
    MsgBox Err.Description
    Resume Exit_Command5_Click
End Sub

Private Sub Command6_Click()
On Error GoTo Err_Command6_Click
    Dim stDocName As String
    stDocName = "Precard"
    DoCmd.OpenReport stDocName, acNormal, , "[code]= form![Qcard2].[code]"
Exit_Command6_Click:
    Exit Sub
Err_Command6_Click:
    MsgBox Err.Description
    Resume Exit_Command6_Click
End Sub

Private Sub Command7_Click()
On Error GoTo Err_Command7_Click
    Dim stDocName As String
```

```
    stDocName = "Precard"  
    DoCmd.OpenReport stDocName, acPreview, , "[code]= forms![Qcard2].[code]"  
Exit_Command7_Click:  
    Exit Sub  
Err_Command7_Click:  
    MsgBox Err.Description  
    Resume Exit_Command7_Click  
  
End Sub  
  
Private Sub Command18_Click()  
On Error GoTo Err_Command18_Click  
    Dim stDocName As String  
    stDocName = "Qcard"  
    DoCmd.OpenReport stDocName, acNormal  
Exit_Command18_Click:  
    Exit Sub  
Err_Command18_Click:  
    MsgBox Err.Description  
    Resume Exit_Command18_Click  
End Sub  
  
Private Sub Command19_Click()  
On Error GoTo Err_Command19_Click  
    Dim stDocName As String  
    stDocName = "Precard"  
    DoCmd.OpenReport stDocName, acNormal  
Exit_Command19_Click:  
    Exit Sub  
Err_Command19_Click:  
    MsgBox Err.Description
```

```

    Resume Exit_Command19_Click
End Sub

Private Sub Command20_Click()
On Error GoTo Err_Command20_Click
    Dim stDocName As String
    stDocName = "Qcard"
    DoCmd.OpenReport stDocName, acNormal, , "[code] =forms![Patient].[code]"
Exit_Command20_Click:
    Exit Sub
Err_Command20_Click:
    MsgBox Err.Description
    Resume Exit_Command20_Click
End Sub

Private Sub Command22_Click()
On Error GoTo Err_Command22_Click
    DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
    DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
Exit_Command22_Click:
    Exit Sub
Err_Command22_Click:
    MsgBox Err.Description
    Resume Exit_Command22_Click
End Sub

```

3.19 คลาสโมดูลของ Form SeaCome(No)

```

Private Sub Command10_Click()

```

```
On Error GoTo Err_Command10_Click
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
Exit_Command10_Click:
    Exit Sub
Err_Command10_Click:
    MsgBox Err.Description
    Resume Exit_Command10_Click
End Sub

Private Sub Command12_Click()
On Error GoTo Err_Command12_Click
    DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
    DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
Exit_Command12_Click:
    Exit Sub
Err_Command12_Click:
    MsgBox Err.Description
    Resume Exit_Command12_Click
End Sub

Private Sub Command15_Click()
On Error GoTo Err_Command15_Click
    DoCmd.Close
Exit_Command15_Click:
    Exit Sub
Err_Command15_Click:
    MsgBox Err.Description
    Resume Exit_Command15_Click
End Sub

Private Sub Form_Open(Cancel As Integer)
```

```

Dim stDocName As String
stDocName = "SearchComein(name)"
DoCmd.OpenForm stDocName, acNormal, , "[name] = [พิมพ์ชื่อ-สกุลผู้ป่วยที่ต้องการหา]"
Exit_Command79_Click:
    Exit Sub
Err_Command79_Click:
    MsgBox Err.Description
    Resume Exit_Command79_Click
End Sub

```

```

Private Sub Command17_Click()
On Error GoTo Err_Command17_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "Comein1"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command17_Click:
    Exit Sub
Err_Command17_Click:
    MsgBox Err.Description
    Resume Exit_Command17_Click

End Sub

```

```

Private Sub Command27_Click()
On Error GoTo Err_Command27_Click
    Screen.PreviousControl.SetFocus
    DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70
Exit_Command27_Click:
    Exit Sub
Err_Command27_Click:

```



```

MsgBox Err.Description
Resume Exit_Command27_Click
End Sub

```

3.20 คลาสโมดูลของ Form SearchComein

```

Private Sub Command10_Click()
On Error GoTo Err_Command10_Click
DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
Exit_Command10_Click:
Exit Sub
Err_Command10_Click:
MsgBox Err.Description
Resume Exit_Command10_Click
End Sub

```

```

Private Sub Command12_Click()
On Error GoTo Err_Command12_Click
DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
Exit_Command12_Click:
Exit Sub
Err_Command12_Click:
MsgBox Err.Description
Resume Exit_Command12_Click

End Sub

```

```

Private Sub Command15_Click()
On Error GoTo Err_Command15_Click
DoCmd.Close

```

```
Exit_Command15_Click:
```

```
Exit Sub
```

```
Err_Command15_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_Command15_Click
```

```
End Sub
```

```
Private Sub Form_Open(Cancel As Integer)
```

```
Dim stDocName As String
```

```
stDocName = "SearchComein"
```

```
DoCmd.OpenForm stDocName, acNormal, , "[code] = [กรอกเลขประจำตัวผู้ปวย]"
```

```
If IsNull(code) = 0 Then
```

```
Else
```

```
MsgBox "ไม่มีข้อมูลในระบบ", 0, "แจ้งให้ทราบ"
```

```
DoCmd.Close
```

```
End If
```

```
End Sub
```

```
Private Sub Command17_Click()
```

```
On Error GoTo Err_Command17_Click
```

```
Dim stDocName As String
```

```
Dim stLinkCriteria As String
```

```
stDocName = "Comein1"
```

```
DoCmd.OpenForm stDocName, , , stLinkCriteria
```

```
Exit_Command17_Click:
```

```
Exit Sub
```

```
Err_Command17_Click:
```

```
MsgBox Err.Description
```

```
Resume Exit_Command17_Click
```

```
End Sub
```

3.21 คลาสโมดูลของ Form SearchComein(name)

```
Private Sub Command10_Click()
On Error GoTo Err_Command10_Click
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
Exit_Command10_Click:
    Exit Sub
Err_Command10_Click:
    MsgBox Err.Description
    Resume Exit_Command10_Click
End Sub
```

```
Private Sub Command12_Click()
On Error GoTo Err_Command12_Click
    DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
    DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
Exit_Command12_Click:
    Exit Sub
Err_Command12_Click:
    MsgBox Err.Description
    Resume Exit_Command12_Click
End Sub
```

```
Private Sub Command15_Click()
On Error GoTo Err_Command15_Click
    DoCmd.Close
Exit_Command15_Click:
    Exit Sub
Err_Command15_Click:
    MsgBox Err.Description
```

```
Resume Exit_Command15_Click
End Sub

Private Sub Form_Open(Cancel As Integer)
Dim stDocName As String
stDocName = "SearchComein(name)"
DoCmd.OpenForm stDocName, acNormal, , "[name] = [พิมพ์ชื่อ-สกุลผู้ป่วยที่ต้องการหา]"
End Sub

Private Sub Command17_Click()
On Error GoTo Err_Command17_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "Comein1"
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Command17_Click:
Exit Sub
Err_Command17_Click:
MsgBox Err.Description
Resume Exit_Command17_Click
End Sub
```

4. แมโคร(Macro)

4.1 AddComein

OpenForm	จากฟอร์ม Comein1 , โหมดข้อมูลอ่านอย่างเดียว
AddMenu	เมนูชื่อแมโคร menu
AddMenu	เมนูชื่อแมโคร menuadd
AddMenu	เมนูชื่อแมโคร edit
AddMenu	เมนูชื่อแมโคร search
AddMenu	เมนูชื่อแมโคร print

4.2 AddDoc

OpenForm	จากฟอร์ม AddDoctors , โหมดข้อมูลเพิ่ม
AddMenu	เมนูชื่อแมโคร menu
AddMenu	เมนูชื่อแมโคร menuadd
AddMenu	เมนูชื่อแมโคร edit
AddMenu	เมนูชื่อแมโคร search
AddMenu	เมนูชื่อแมโคร print

4.3 Addnew

OpenForm	จากฟอร์ม Patientnew , โหมดข้อมูลแก้ไข
AddMenu	เมนูชื่อแมโคร menu
AddMenu	เมนูชื่อแมโคร menuadd
AddMenu	เมนูชื่อแมโคร edit
AddMenu	เมนูชื่อแมโคร search
AddMenu	เมนูชื่อแมโคร print

4.4 Edit

OpenForm	จากฟอร์ม EditPatient , โหมดข้อมูลแก้ไข
----------	--

OpenForm	จากฟอร์ม EditDoctors , โหมดข้อมูลแก้ไข
OpenForm	จากฟอร์ม Comein1, โหมดข้อมูลแก้ไข

4.5 EditDoc

OpenForm	จากฟอร์ม EditDoctors , โหมดข้อมูลแก้ไข
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit
AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.6 Editpat

OpenForm	จากฟอร์ม EditPatient , โหมดข้อมูลแก้ไข
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit
AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.7 Mainmenu

OpenForm	จากฟอร์ม Main , โหมดข้อมูลอ่านอย่างเดียว
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit
AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.8 Menu

OpenForm	จากฟอร์ม Main , โหมดข้อมูลอ่านอย่างเดียว
OpenQuery	ชื่อแบบสอบถาม AllPatient , โหมดข้อมูลแก้ไข

OpenQuary	ชื่อแบบสอบถาม ComeinNow , โหมดข้อมูลแก้ไข
OpenQuary	ชื่อแบบสอบถาม AllComein , โหมดข้อมูลอ่านอย่างเดียว
OpenQuary	ชื่อแบบสอบถาม Doctors, โหมดข้อมูลแก้ไข
OpenQuary	ชื่อแบบสอบถาม AllQ , โหมดข้อมูลแก้ไข
MsgBox	ชนิดการเตือน(ออกจากโปรแกรม)
Quit	ออกจากโปรแกรม

4.9 Menuadd

OpenForm	จากฟอร์ม PatientNew , โหมดข้อมูลแก้ไข
OpenForm	จากฟอร์ม AddDoctors , โหมดข้อมูลเพิ่ม
OpenForm	จากฟอร์ม Comein1 , โหมดข้อมูลแก้ไข

4.10 MenuDoc

OpenQuary	จากแบบสอบถาม Doctors , โหมดข้อมูลแก้ไข
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit
AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.11 MenuPat

OpenQuary	จากแบบสอบถาม AllPatient , โหมดข้อมูลแก้ไข
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit
AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.12 Patientfrm

OpenForm	จากฟอร์ม Patient , โหมดข้อมูลอ่านอย่างเดียว
AddMenu	เมนูชื่อแม่โคร menu

AddMenu	เมนูชื่อแม่โคร	menuadd
AddMenu	เมนูชื่อแม่โคร	edit
AddMenu	เมนูชื่อแม่โคร	search
AddMenu	เมนูชื่อแม่โคร	print

4.13 Patientfrm2

OpenForm	จากฟอร์ม Patient(name) , โหมดข้อมูลอ่านอย่างเดียว
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit
AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.14 Patientfrm3

OpenForm	จากฟอร์ม Patient(No) , โหมดข้อมูลอ่านอย่างเดียว
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit
AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.15 Print

MsgBox	พร้อมที่จะพิมพ์
PrintOut	พิมพ์

4.16 PrintPreview

OpenReport	จากรายงาน AboutPatient
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit

AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.17 Qcard

OpenForm	จากฟอร์ม Qcard , โหมดข้อมูลแก้ไข
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit
AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.18 Room

OpenForm	จากฟอร์ม Comein3 , โหมดข้อมูลแก้ไข
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit
AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.19 SeaCome

OpenForm	จากฟอร์ม SeaCome(No) , โหมดข้อมูลอ่านอย่างเดียว
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit
AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.20 Search

OpenForm	จากฟอร์ม Patien(No) , โหมดข้อมูลอ่านอย่างเดียว
OpenForm	จากฟอร์ม SearchComein , โหมดข้อมูลอ่านอย่างเดียว
OpenForm	จากฟอร์ม SearchComein(name) , โหมดข้อมูลอ่านอย่างเดียว

OpenForm	จากฟอร์ม SeaCome(No) , โหมดข้อมูลอ่านอย่างเดียว
OpenForm	จากฟอร์ม Doctor2 , โหมดข้อมูลแก้ไข

4.21 SearchComein

OpenForm	จากฟอร์ม SearchComein , โหมดข้อมูลอ่านอย่างเดียว
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit
AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.22 SeachComein(name)

OpenForm	จากฟอร์ม SearchComein(name) , โหมดข้อมูลเพิ่ม
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit
AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.23 SearchDoc

OpenForm	จากฟอร์ม Doctors , โหมดข้อมูลอ่านอย่างเดียว
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd
AddMenu	เมนูชื่อแม่โคร edit
AddMenu	เมนูชื่อแม่โคร search
AddMenu	เมนูชื่อแม่โคร print

4.24 SearchDoc2

OpenForm	จากฟอร์ม Doctor2 , โหมดข้อมูลแก้ไข
AddMenu	เมนูชื่อแม่โคร menu
AddMenu	เมนูชื่อแม่โคร menuadd

AddMenu	เมนูชื่อแม่โคร	edit
AddMenu	เมนูชื่อแม่โคร	search
AddMenu	เมนูชื่อแม่โคร	print

4.25 SearchPatient

OpenForm	จากฟอร์ม Patient , โหมดข้อมูลอ่านอย่างเดียว	
AddMenu	เมนูชื่อแม่โคร	menu
AddMenu	เมนูชื่อแม่โคร	menuadd
AddMenu	เมนูชื่อแม่โคร	edit
AddMenu	เมนูชื่อแม่โคร	search
AddMenu	เมนูชื่อแม่โคร	print

โปรแกรมภาษาแอสเซมบลี (Assembly) ซึ่งใช้ในการควบคุมรถขนส่งอัตโนมัติแสดงได้ดังนี้

```
*****
;
; PROGRAM      :AUTOMATIC TRANSPORT CONTROL
; DESCRIPTION  :KEYPAD INPUT AND SHOW KEY VOLUME TO LCD MODULE
;
;              :ROUTE DETECT BY INFRARED SENSOR FOR AUTOMATIC
;              TRANSPORT
; FOR          : AT89S53
*****
;
*****
; DEFINE PORT AND PIN NAME
*****
;
        ROW_0          BIT          P1.0
        ROW_1          BIT          P1.1
        ROW_2          BIT          P1.2
        ROW_3          BIT          P1.3
        COL_3          BIT          P1.4
        COL_2          BIT          P1.5
        COL_1          BIT          P1.6
        COL_0          BIT          P1.7
        LCD_EN         BIT          P3.6
        LCD_RS         BIT          P3.7
*****
; DEFINE USER REGISTER
*****
;
        LCD_ADDR       EQU          030H
        LCD_DATA       EQU          031H
        LCD_PTR        EQU          032H
```

```
KPAD_DATA      EQU      033H
START_POINT    EQU      034H
END_POINT      EQU      035H
```

```
*****
;
```

```
; MAIN PROGRAM
```

```
*****
;
```

```
          ORG          0000H
          AJMP         INITIAL

INITIAL:  MOV          P0 , #00000000B
          MOV          P1 , #11111111B
          MOV          P2 , #00000000B
          MOV          P3 , #00011111B

MAIN:    CALL          INIT_LCD
LOOP:    MOV          LCD_ADDR , #000H
          CALL         SET_ADDR_LCD
          MOV          DPTR , #TITLE_1
          CALL         WRLINE_LCD
          CALL         DELAY_1S
          CALL         DELAY_1S
          MOV          LCD_ADDR , #000H
          CALL         SET_ADDR_LCD
          MOV          DPTR , #TITLE_2
          CALL         WRLINE_LCD
          CALL         DELAY_1S
          CALL         DELAY_1S
```

```

BEGIN:      MOV      LCD_ADDR , #000H
            CALL     SET_ADDR_LCD
            MOV      DPTR , #FROM
            CALL     WRLINE_LCD
            CALL     DELAY_1S
            CALL     INIT_LCD
            MOV      LCD_ADDR , #000H
            CALL     SET_ADDR_LCD
            CALL     GET_KEYPAD
            MOV      A , KPAD_DATA

DSP_0:      CJNE     A , #14 , DSP_1
            MOV      DPTR,#FROM_0
            CALL     WRLINE_LCD
            CALL     DELAY_1S
            CALL     DELAY_1S
            LJMP     START_0

DSP_1:      CJNE     A , #1 , DSP_2
            MOV      DPTR , #FROM_1
            CALL     WRLINE_LCD
            CALL     DELAY_1S
            CALL     DELAY_1S
            LJMP     START_1

DSP_2:      CJNE     A , #2 , DSP_3
            MOV      DPTR,#FROM_2
            CALL     WRLINE_LCD
            CALL     DELAY_1S

```

```

                CALL        DELAY_1S
                LJMP        START_2

DSP_3:         CJNE        A , #3 , DSP_4
                MOV        DPTR , #FROM_3
                CALL        WRLINE_LCD
                CALL        DELAY_1S
                CALL        DELAY_1S
                LJMP        START_3

DSP_4:         CJNE        A , #5 , DSP_CLR
                MOV        DPTR , #FROM_4
                CALL        WRLINE_LCD
                CALL        DELAY_1S
                CALL        DELAY_1S
                LJMP        START_4

DSP_CLR:      CJNE        A , #13 , WAIT
                LJMP        INITIAL

```

```

,*****
,

```

```

; START FROM STATION 0

```

```

,*****
,

```

```

START_0:      CALL        INIT_LCD
                MOV        LCD_ADDR , #000H
                CALL        SET_ADDR_LCD
                MOV        DPTR , #SEND_TO
                CALL        WRLINE_LCD
                CALL        DELAY_1S

```

```

MOV      LCD_ADDR , #000H
CALL     SET_ADDR_LCD
CALL     GET_KEYPAD
MOV      A , KPAD_DATA

PRESS_1: CJNE      A , #1 , PRESS_2
MOV      DPTR , #FROM_0TO1
CALL     WRLINE_LCD
CALL     DELAY_1S
LJMP    F0TO1

PRESS_2: CJNE      A , #2 , PRESS_3
MOV      DPTR,#FROM_0TO2
CALL     WRLINE_LCD
CALL     DELAY_1S
LJMP    F0TO2

PRESS_3: CJNE      A , #3 , PRESS_4
MOV      DPTR , #FROM_0TO3
CALL     WRLINE_LCD
CALL     DELAY_1S
LJMP    F0TO3

PRESS_4: CJNE      A , #5 , PRESS_CLR
MOV      DPTR , #FROM_0TO4
CALL     WRLINE_LCD
CALL     DELAY_1S
LJMP    F0TO4

```



```
PRESS_CLR:      CJNE          A , #13 , WAIT
                LJMP          INITIAL
```

```
WAIT:          CALL          WAIT_KEY
                LJMP          INITIAL
```

```
,*****
;
```

```
; START FROM STATION 1
```

```
,*****
;
```

```
START_1:      CALL          INIT_LCD
                MOV           LCD_ADDR , #000H
                CALL         SET_ADDR_LCD
                MOV           DPTR , #SEND_TO
                CALL         WRLINE_LCD
                CALL         DELAY_1S
                MOV           LCD_ADDR , #000H
                CALL         SET_ADDR_LCD
                CALL         GET_KEYPAD
                MOV           A , KPAD_DATA
```

```
XP_1:        CJNE          A , #14 , XP_2
                MOV           DPTR , #FROM_1TO0
                CALL         WRLINE_LCD
                CALL         DELAY_1S
                LJMP         F1TO0
```

```
XP_2:        CJNE          A , #2 , XP_3
                MOV           DPTR , #FROM_1TO2
                CALL         WRLINE_LCD
                CALL         DELAY_1S
                LJMP         F1TO2
```

```

XP_3:          CJNE          A,#3,XP_4
               MOV          DPTR,#FROM_1TO3
               CALL         WRLINE_LCD
               CALL         DELAY_1S
               LJMP         F1TO3

```

```

XP_4:          CJNE          A,#5,XP_CLR
               MOV          DPTR,#FROM_1TO4
               CALL         WRLINE_LCD
               CALL         DELAY_1S
               LJMP         F1TO4

```

```

XP_CLR:       CJNE          A,#13,WAIT
               LJMP         INITIAL

```

```

,*****
,

```

```

; FROM STATION 2

```

```

,*****
,

```

```

START_2:      CALL         INIT_LCD
               MOV          LCD_ADDR,#000H
               CALL         SET_ADDR_LCD
               MOV          DPTR,#SEND_TO
               CALL         WRLINE_LCD
               CALL         DELAY_1S
               MOV          LCD_ADDR,#000H
               CALL         SET_ADDR_LCD
               CALL         GET_KEYPAD
               MOV          A,KPAD_DATA

```

```

SS_1:        CJNE          A,#14,SS_2

```

```

MOV      DPTR,# FROM_2TO0
CALL     WRLINE_LCD
CALL     DELAY_1S
LJMP     F2TO0

SS_2:    CJNE      A,#1,SS_3
MOV      DPTR,#FROM_2TO1
CALL     WRLINE_LCD
CALL     DELAY_1S
LJMP     F2TO1

SS_3:    CJNE      A,#3,SS_4
MOV      DPTR,#FROM_2TO3
CALL     WRLINE_LCD
CALL     DELAY_1S
LJMP     F2TO3

SS_4:    CJNE      A,#5,WAIT
MOV      DPTR,#FROM_2TO4
CALL     WRLINE_LCD
CALL     DELAY_1S
LJMP     F2TO4

;*****
;
; FROM STATION 3
;*****
START_3: CALL     INIT_LCD
MOV      LCD_ADDR,#000H
CALL     SET_ADDR_LCD

```

```

MOV      DPTR,#SEND_TO
CALL     WRLINE_LCD
CALL     DELAY_1S
MOV      LCD_ADDR,#000H
CALL     SET_ADDR_LCD
CALL     GET_KEYPAD
MOV      A,KPAD_DATA

S_1:     CJNE      A,#14,S_2
MOV      DPTR,#FROM_3TO0
CALL     WRLINE_LCD
CALL     DELAY_1S
LJMP     F3TO0

S_2:     CJNE      A,#1,S_3
MOV      DPTR,#FROM_3TO1
CALL     WRLINE_LCD
CALL     DELAY_1S
LJMP     F3TO1

S_3:     CJNE      A,#2,S_4
MOV      DPTR,#FROM_3TO2
CALL     WRLINE_LCD
CALL     DELAY_1S
LJMP     F3TO2

S_4:     CJNE      A,#5,S_CLR
MOV      DPTR,#FROM_3TO4
CALL     WRLINE_LCD

```

```

                                CALL    DELAY_1S
                                LJMP    F3T04

S_CLR:                          CJNE    A,#13,WAIT
                                LJMP    INITIAL

```

```

,*****
; FROM STATION 4
,*****

```

```

START_4:                        CALL    INIT_LCD
                                MOV     LCD_ADDR,#000H
                                CALL    SET_ADDR_LCD
                                MOV     DPTR,#SEND_TO
                                CALL    WRLINE_LCD
                                CALL    DELAY_1S
                                MOV     LCD_ADDR,#000H
                                CALL    SET_ADDR_LCD
                                CALL    GET_KEYPAD
                                MOV     A,KPAD_DATA

```

```

SSS_1:                          CJNE    A,#14,SSS_2
                                MOV     DPTR,#FROM_4T00
                                CALL    WRLINE_LCD
                                CALL    DELAY_1S
                                LJMP    F4T00

```

```
SSS_2:          CJNE          A,#1,SSS_3
                MOV          DPTR,#FROM_4TO1
                CALL         WRLINE_LCD
                CALL         DELAY_1S
                LJMP         F4TO1
```

```
SSS_3:          CJNE          A,#2,SSS_4
                MOV          DPTR,#FROM_4TO2
                CALL         WRLINE_LCD
                CALL         DELAY_1S
                LJMP         F4TO2
```

```
SSS_4:          CJNE          A,#3,SSS_CLR
                MOV          DPTR,#FROM_4TO3
                CALL         WRLINE_LCD
                CALL         DELAY_1S
                LJMP         F4TO3
```

```
SSS_CLR:        CJNE  A,#13,WAIT
                LJMP  INITIAL
```

```
*****
;
; FROM STATION 0 TO STATION 1
*****
;
```

```
F0TO1:          CALL         INIT_LCD
                MOV          LCD_ADDR,#000H
                CALL         SET_ADDR_LCD
```

```

MOV      DPTR,#FROM_0TO1
CALL    WRLINE_LCD
CALL    DELAY_100MS
MOV     LCD_ADDR,#000H
CALL    SET_ADDR_LCD
CALL    GET_KEYPAD
MOV     A,KPAD_DATA

```

```

A_1:    CJNE      A,#16,A_2
        LJMP     GO_0TO1

```

```

A_2:    CJNE      A,#13,WAIT
        LJMP     INITIAL

```

```

,*****
,

```

```

;FROM STATION 0 TO STATION 2

```

```

,*****
,

```

```

F0TO2:  CALL      INIT_LCD
        MOV     LCD_ADDR,#000H
        CALL    SET_ADDR_LCD
        MOV     DPTR,#FROM_0TO2
        CALL    WRLINE_LCD
        CALL    DELAY_100MS
        MOV     LCD_ADDR,#000H
        CALL    SET_ADDR_LCD
        CALL    GET_KEYPAD
        MOV     A,KPAD_DATA

```

```
B_1:          CJNE          A,#16,B_2
              LJMP          GO_0TO2
```

```
B_2:          CJNE          A,#13,WAIT
              LJMP          INITIAL
```

```
*****
; FROM STATION 0 TO STATION 3
*****
```

```
F0TO3:       CALL          INIT_LCD
              MOV           LCD_ADDR,#000H
              CALL          SET_ADDR_LCD
              MOV           DPTR,#FROM_0TO3
              CALL          WRLINE_LCD
              CALL          DELAY_100MS
              MOV           LCD_ADDR,#000H
              CALL          SET_ADDR_LCD
              CALL          GET_KEYPAD
              MOV           A,KPAD_DATA
```

```
C_1:          CJNE          A,#16,C_2
              LJMP          GO_0TO3
```

```
C_2:          CJNE          A,#13,WAIT
              LJMP          INITIAL
```

```
*****
; FROM STATION 0 TO STATION 4
*****
```



```
.*****  
,
```

```
F0T04:      CALL      INIT_LCD  
            MOV       LCD_ADDR,#000H  
            CALL     SET_ADDR_LCD  
            MOV       DPTR,#FROM_0T04  
            CALL     WRLINE_LCD  
            CALL     DELAY_100MS  
            MOV       LCD_ADDR,#000H  
            CALL     SET_ADDR_LCD  
            CALL     GET_KEYPAD  
            MOV       A,KPAD_DATA
```

```
D_1:        CJNE      A,#16,D_2  
            LJMP     GO_0T04
```

```
D_2:        CJNE      A,#13,WAIT  
            LJMP     INITIAL
```

```
.*****  
,
```

```
; FROM STATION 1 TO STATION 0
```

```
.*****  
,
```

```
F1T00:      CALL      INIT_LCD  
            MOV       LCD_ADDR,#000H  
            CALL     SET_ADDR_LCD  
            MOV       DPTR,#FROM_1T00  
            CALL     WRLINE_LCD  
            CALL     DELAY_100MS  
            MOV       LCD_ADDR,#000H  
            CALL     SET_ADDR_LCD
```

```

                                CALL      GET_KEYPAD
                                MOV       A,KPAD_DATA

E_1:                            CJNE     A,#16,E_2
                                LJMP     GO_1TO0

E_2:                            CJNE     A,#13,WAIT
                                LJMP     INITIAL

,*****
,
; FROM STATION 1 TO STATION 2
,*****
,

F1TO2:                          CALL     INIT_LCD
                                MOV      LCD_ADDR,#000H
                                CALL     SET_ADDR_LCD
                                MOV      DPTR,#FROM_1TO2
                                CALL     WRLINE_LCD
                                CALL     DELAY_100MS
                                MOV      LCD_ADDR,#000H
                                CALL     SET_ADDR_LCD
                                CALL     GET_KEYPAD
                                MOV      A,KPAD_DATA

F_1:                            CJNE     A,#16,F_2
                                LJMP     GO_1TO2

F_2:                            CJNE     A,#13,WAIT
                                LJMP     INITIAL

```

.*****
,

; FROM STATION 1 TO STATION 3

.*****
,

```
F1TO3:          CALL      INIT_LCD
                MOV       LCD_ADDR,#000H
                CALL      SET_ADDR_LCD
                MOV       DPTR,#FROM_1TO3
                CALL      WRLINE_LCD
                CALL      DELAY_100MS
                MOV       LCD_ADDR,#000H
                CALL      SET_ADDR_LCD
                CALL      GET_KEYPAD
                MOV       A,KPAD_DATA
```

```
G_1:           CJNE      A,#16,G_2
                LJMP      GO_1TO3
                MOV       LCD_ADDR,#000H
                CALL      SET_ADDR_LCD
                MOV       DPTR,#WORK
                CALL      WRLINE_LCD
                CALL      DELAY_1S
                CALL      DELAY_1S
```

```
G_2:           CJNE      A,#13,WAIT
                LJMP      INITIAL
```

.*****
,

; FROM STATION 1 TO STATION 4

.*****
,

```

F1TO4:          CALL      INIT_LCD

                MOV       LCD_ADDR,#000H
                CALL     SET_ADDR_LCD
                MOV       DPTR,#FROM_1TO4
                CALL     WRLINE_LCD
                CALL     DELAY_100MS
                MOV       LCD_ADDR,#000H
                CALL     SET_ADDR_LCD
                CALL     GET_KEYPAD
                MOV       A,KPAD_DATA

```

```

H_1:           CJNE      A,#16,H_2
                LJMP     GO_1TO4

```

```

H_2:           CJNE      A,#13,WAIT
                LJMP     INITIAL

```

```

,*****
,
; FROM STATION 2 TO STATION 0

```

```

,*****
,

```

```

F2TO0:          CALL      INIT_LCD

                MOV       LCD_ADDR,#000H
                CALL     SET_ADDR_LCD
                MOV       DPTR,#FROM_2TO0
                CALL     WRLINE_LCD
                CALL     DELAY_100MS
                MOV       LCD_ADDR,#000H
                CALL     SET_ADDR_LCD
                CALL     GET_KEYPAD

```

```

MOV      A,KPAD_DATA

I_1:    CJNE      A,#16,I_2
        LJMP      GO_2TO0

I_2:    CJNE      A,#13,WAIT
        LJMP      INITIAL

;*****
;
; FROM STATION 2 TO STATION 1
;*****
;

F2TO1:  CALL      INIT_LCD
        MOV      LCD_ADDR,#000H
        CALL     SET_ADDR_LCD
        MOV      DPTR,#FROM_2TO1
        CALL     WRLINE_LCD
        CALL     DELAY_100MS

        MOV      LCD_ADDR,#000H
        CALL     SET_ADDR_LCD
        CALL     GET_KEYPAD
        MOV      A,KPAD_DATA

J_1:    CJNE      A,#16,J_2
        LJMP      GO_2TO1

J_2:    CJNE      A,#13,WAIT
        LJMP      INITIAL

```

.*****
,

; FROM STATION 2 TO STATION 3

.*****
,

```
F2TO3:          CALL      INIT_LCD
                MOV       LCD_ADDR,#000H
                CALL      SET_ADDR_LCD
                MOV       DPTR,#FROM_2TO3
                CALL      WRLINE_LCD
                CALL      DELAY_100MS
                MOV       LCD_ADDR,#000H
                CALL      SET_ADDR_LCD
                CALL      GET_KEYPAD
                MOV       A,KPAD_DATA
```

```
K_1:           CJNE      A,#16,K_2
                LJMP      GO_2TO3
```

```
K_2:           CJNE      A,#13,WAIT
                LJMP      INITIAL
```

.*****
,

; FROM STATION 2 TO STATION 4

.*****
,

```
F2TO4:          CALL      INIT_LCD
                MOV       LCD_ADDR,#000H
                CALL      SET_ADDR_LCD
                MOV       DPTR,#FROM_2TO4
                CALL      WRLINE_LCD
                CALL      DELAY_100MS
                MOV       LCD_ADDR,#000H
```

```

CALL      SET_ADDR_LCD
CALL      GET_KEYPAD
MOV       A,KPAD_DATA

L_1:      CJNE      A,#16,L_2
          LJMP      GO_2TO4

L_2:      CJNE      A,#13,WAIT
          LJMP      INITIAL

```

```

,*****
,
; FROM STATION 3 TO STATION 0
,*****
,

```

```

F3TO0:    CALL      INIT_LCD
          MOV       LCD_ADDR,#000H
          CALL      SET_ADDR_LCD
          MOV       DPTR,#FROM_3TO0
          CALL      WRLINE_LCD
          CALL      DELAY_100MS

          MOV       LCD_ADDR,#000H
          CALL      SET_ADDR_LCD
          CALL      GET_KEYPAD
          MOV       A,KPAD_DATA

```

```

M_1:      CJNE      A,#16,M_2
          LJMP      GO_3TO0

```

```

M_2:      CJNE      A,#13,WAIT

```

LJMP INITIAL

,

; FROM STATION 3 TO STATION 1

,

F3TO1: CALL INIT_LCD
MOV LCD_ADDR,#000H
CALL SET_ADDR_LCD
MOV DPTR,#FROM_3TO1
CALL WRLINE_LCD
CALL DELAY_1S
CALL DELAY_1S
MOV LCD_ADDR,#000H
CALL SET_ADDR_LCD
CALL GET_KEYPAD
MOV A,KPAD_DATA

N_1: CJNE A,#16,N_2
LJMP GO_3TO1

N_2: CJNE A,#13,WAIT
LJMP INITIAL

,

; FROM STATION 3 TO STATION 2

,

F3TO2: CALL INIT_LCD
MOV LCD_ADDR,#000H


```

CALL      SET_ADDR_LCD
MOV       DPTR,#FROM_3TO2
CALL     WRLINE_LCD
CALL     DELAY_1S
CALL     DELAY_1S
MOV      LCD_ADDR,#000H
CALL     SET_ADDR_LCD
CALL     GET_KEYPAD
MOV      A,KPAD_DATA

O_1:     CJNE      A,#16,O_2
         LJMP     GO_3TO2

O_2:     CJNE      A,#13,WAIT
         LJMP     INITIAL

```

```

;*****
;
; FROM STATION 3 TO STATION 4
;*****
;

```

```

F3TO4:  CALL      INIT_LCD
         MOV      LCD_ADDR,#000H
         CALL     SET_ADDR_LCD
         MOV      DPTR,#FROM_3TO4
         CALL     WRLINE_LCD
         CALL     DELAY_1S
         CALL     DELAY_1S
         MOV      LCD_ADDR,#000H
         CALL     SET_ADDR_LCD
         CALL     GET_KEYPAD

```

```

MOV      A,KPAD_DATA

X_1:    CJNE      A,#16,X_2
        LJMP      GO_3TO4

X_2:    CJNE      A,#13,WAIT
        LJMP      INITIAL

;*****
;
; FROM STATION 4 TO STATION 0
;*****
;
F4TO0:  CALL      INIT_LCD
        MOV      LCD_ADDR,#000H
        CALL     SET_ADDR_LCD
        MOV      DPTR,#FROM_4TO0
        CALL     WRLINE_LCD
        CALL     DELAY_100MS
        MOV      LCD_ADDR,#000H
        CALL     SET_ADDR_LCD
        CALL     GET_KEYPAD
        MOV      A,KPAD_DATA

Q_1:    CJNE      A,#16,Q_2
        LJMP      GO_4TO0

Q_2:    CJNE      A,#13,WAIT
        LJMP      INITIAL

;*****
;

```

; FROM STATION 4 TO STATION 1

,

```
F4TO1:      CALL      INIT_LCD
            MOV      LCD_ADDR,#000H
            CALL     SET_ADDR_LCD
            MOV      DPTR,#FROM_4TO1
            CALL     WRLINE_LCD
            CALL     DELAY_100MS
            MOV      LCD_ADDR,#000H
            CALL     SET_ADDR_LCD
            CALL     GET_KEYPAD
            MOV      A,KPAD_DATA
```

```
R_1:        CJNE      A,#16,R_2
            LJMP     GO_4TO1
```

```
R_2:        CJNE      A,#13,WAIT
            LJMP     INITIAL
```

,

; FROM STATION 4 TO STATION 2

,

```
F4TO2:      CALL      INIT_LCD
            MOV      LCD_ADDR,#000H
            CALL     SET_ADDR_LCD
            MOV      DPTR,#FROM_4TO2
            CALL     WRLINE_LCD
            CALL     DELAY_100MS
            MOV      LCD_ADDR,#000H
```

```

CALL      SET_ADDR_LCD
CALL      GET_KEYPAD
MOV       A,KPAD_DATA

Y_1:      CJNE      A,#16,Y_2
          LJMP      GO_4TO2

Y_2:      CJNE      A,#13,WAIT
          LJMP      INITIAL

```

```

,*****
,
; FROM STATION 4 TO STATION 3
,*****
,

```

```

F4TO3:    CALL      INIT_LCD
          MOV       LCD_ADDR,#000H
          CALL      SET_ADDR_LCD
          MOV       DPTR,#FROM_4TO3
          CALL      WRLINE_LCD
          CALL      DELAY_100MS
          MOV       LCD_ADDR,#000H
          CALL      SET_ADDR_LCD
          CALL      GET_KEYPAD
          MOV       A,KPAD_DATA

```

```

T_1:      CJNE      A,#16,T_2
          LJMP      GO_4TO3

T_2:      CJNE      A,#13,WAIT
          LJMP      INITIAL

```

```
.*****  
,  
; DISPLAY 'ERROR'  
.*****  
,
```

```
DSP_ERROR:      MOV      LCD_ADDR,#000H  
                CALL     SET_ADDR_LCD  
                MOV      DPTR,#ERROR  
                CALL     WRLINE_LCD  
                CALL     DELAY_1S  
                CALL     DELAY_1S  
                LJMP     INITIAL
```

```
.*****  
,  
; LCD INITIALIZE  
.*****  
,
```

```
INIT_LCD:      CALL     DELAY_100mS  
                CLR      LCD_RS  
                MOV      P0,#00111000B  
                CALL     LCD_CLK  
                CALL     DELAY_10mS  
                MOV      P0,#00111000B  
                CALL     LCD_CLK  
                CALL     LCD_OFF  
                CALL     LCD_CLR  
  
                MOV      P0,#00000110B  
                CALL     LCD_CLK  
                CALL     LCD_HOME  
                RET
```

```

,*****
;

; LCD CLEAR DISPLAY

,*****

LCD_CLR:          CLR          LCD_RS

                MOV          P0,#00000001B

                CALL         LCD_CLK

                RET

,*****

; LCD DISPLAY OFF

,*****

LCD_OFF:         CLR          LCD_RS

                MOV          P0,#00001000B

                CALL         LCD_CLK

                RET

,*****

; LCD CLK

,*****

LCD_CLK:        SETB         LCD_EN

                CALL         LCD_DELAY

                CLR          LCD_EN

                CALL         LCD_DELAY

                RET

,*****

; LCD DISPLAY ON

,*****

```

```

LCD_ON:          CLR          LCD_RS
                MOV          P0,#00001100B
                CALL         LCD_CLK
                RET

```

```

,*****
,

```

```

;SET LCD ADDRESS

```

```

;/P LCD_ADDR

```

```

,*****
,

```

```

SET_ADDR_LCD:  CLR          LCD_RS
                MOV          A,LCD_ADDR
                SETB         ACC.7
                MOV          P0,A
                CALL         LCD_CLK
                RET

```

```

,*****
,

```

```

;WRITE 16 CHARACTER TO SHOW LCD

```

```

;/P DPTR : LOCATE ROM ADDRESS

```

```

,*****
,

```

```

WRLINE_LCD:    MOV          R0,#0
WRLINE_LCD_1:  SETB         LCD_RS
                CLR          A
                MOVC         A,@A+DPTR
                MOV          P0,A
                CALL         LCD_CLK
                INC          DPTR
                INC          R0
                CJNE         R0,#8,WRLINE_LCD_1

                MOV          LCD_ADDR,#040H

```

```

                                LCALL      SET_ADDR_LCD

WRLINE_LCD_2:  SETB      LCD_RS
                                CLR        A
                                MOVC      A,@A+DPTR
                                MOV        P0,A
                                CALL      LCD_CLK
                                INC        DPTR
                                INC        R0
                                CJNE      R0,#16,WRLINE_LCD_2
                                CALL      LCD_ON
                                RET

;*****
;
; KEYPAD SCANKEY
;*****

GET_KEYPAD:   MOV        P1,#0FFH
                                MOV        KPAD_DATA,#0

CHK_COL_0:   CLR        COL_0
                                MOV        A,P1
                                ANL        A,#00FH
                                CJNE      A,#00FH,COL_0_DETECT
                                LJMP      CHK_COL_1

COL_0_DETECT: MOV        KPAD_DATA,#01
                                LJMP      GET_ROW

CHK_COL_1:   SETB      COL_0
                                CLR        COL_1

```



```

MOV      A,P1
ANL      A,#00FH
CJNE     A,#00FH,COL_1_DETECT
LJMP     CHK_COL_2

COL_1_DETECT:  MOV      KPAD_DATA,#02
                LJMP     GET_ROW

CHK_COL_2:    SETB     COL_1
                CLR      COL_2
                MOV      A,P1
                ANL      A,#00FH
                CJNE     A,#00FH,COL_2_DETECT
                AJMP     CHK_COL_3

COL_2_DETECT:  MOV      KPAD_DATA,#03
                LJMP     GET_ROW

CHK_COL_3:    SETB     COL_2
                CLR      COL_3
                MOV      A,P1
                ANL      A,#00FH
                CJNE     A,#00FH,COL_3_DETECT
                RET

COL_3_DETECT:  MOV      KPAD_DATA,#04

GET_ROW:      CLR      COL_0
                CLR      COL_1

```

```

                CLR        COL_2
                CLR        COL_3
                JB         ROW_0,CHK_ROW_1
                RET

CHK_ROW_1:     JB         ROW_1,CHK_ROW_2
                MOV        A,KPAD_DATA
                ADD        A,#4
                MOV        KPAD_DATA,A
                RET

CHK_ROW_2:     JB         ROW_2,CHK_ROW_3
                MOV        A,KPAD_DATA
                ADD        A,#8
                MOV        KPAD_DATA,A
                RET

CHK_ROW_3:     MOV        A,KPAD_DATA
                ADD        A,#12
                MOV        KPAD_DATA,A
                RET

;*****
;
; WAIT KEYPAD DEPRESSED
;*****
WAIT_KEY:     MOV        A,P1
                ANL        A,#00FH
                CJNE       A,#00FH,WAIT_KEY
                RET

```

```

,*****
,
; DELAY TIME
,*****
,
LCD_DELAY:          MOV          7,#002
LCD_DELAY_1:        MOV          6,#0E6H
LCD_DELAY_2:        NOP
                    NOP
                    DJNZ         R6,LCD_DELAY_2
                    DJNZ         R7,LCD_DELAY_1
                    RET
DELAY_10mS:         MOV          7,#010
DELAY_10mS_1:       MOV          6,#0E6H
DELAY_10mS_2:       NOP
                    NOP
                    DJNZ         R6,DELAY_10mS_2
                    DJNZ         R7,DELAY_10mS_1
                    RET
DELAY_100mS:        MOV          7,#100
DELAY_100mS_1:      MOV          6,#0E6H
DELAY_100mS_2:      NOP
                    NOP
                    DJNZ         R6,DELAY_100mS_2
                    DJNZ         R7,DELAY_100mS_1
                    RET
DELAY_L:            MOV          5,#50
DELAY_B_1:          CALL         DELAY_10MS
                    DJNZ         R5,DELAY_B_1

```



```

RET
TURR:    MOV      P2,#11111010B
RET
TURL:    MOV      P2,#111110101B
RET
STOP:    MOV      P2,#00H
RET
ROTL:    LCALL    TURL
          LCALL    DELAY_1S
          LCALL    DELAY_1S
          RET
ROTR:    LCALL    TURR
          LCALL    DELAY_1S
          LCALL    DELAY_1S
          RET

```

```

,*****
,

```

```

; ROUTE

```

```

,*****
,

```

```

GO_F0TO1: CALL     READ
          CALL     DELAY_L
          CALL     STOP
          CALL     ROTL
          CALL     STOP
          CALL     READ
          CALL     DELAY_L
          CALL     STOP

          CALL     ROTL
          CALL     STOP

```

```
CALL READ
CALL DELAY_L
CALL STOP
CALL ROTL
CALL STOP
CALL READ
CALL DELAY_L
CALL READ
CALL DELAY_L
CALL READ
END
GO_F0TO2: CALL READ
CALL DELAY_L
CALL STOP
CALL ROTL
CALL STOP
CALL READ
CALL DELAY_L
CALL ROTR
CALL STOP
CALL READ
CALL DELAY_L
CALL STOP
END
GO_F0TO3: CALL READ
CALL DELAY_L
CALL STOP
CALL ROTL
CALL STOP
```

```
CALL    READ
CALL    DELAY_L
CALL    STOP
CALL    ROTL
CALL    STOP
CALL    READ
CALL    DELAY_L
CALL    STOP
CALL    ROTR
CALL    STOP
CALL    READ
CALL    DELAY_L
CALL    STOP
CALL    STOP
END
```

```
GO_F0TO4: CALL    READ
CALL    DELAY_L
CALL    STOP
CALL    ROTL
CALL    STOP
CALL    READ
CALL    DELAY_L
CALL    STOP
CALL    DELAY_L
CALL    ROTR
CALL    STOP
CALL    READ
CALL    DELAY_L
```

```
CALL STOP
CALL STOP
END
GO_F1TO0: CALL READ
CALL DELAY_L
CALL STOP
CALL ROTL
CALL STOP
CALL READ
CALL DELAY_L
CALL STOP
CALL ROTL
CALL STOP
CALL READ
CALL DELAY_L
CALL STOP
END
GO_F1TO2: CALL READ
CALL DELAY_L
CALL STOP
CALL ROTL
CALL STOP
CALL READ
CALL DELAY_L
CALL STOP
CALL ROTL
CALL STOP
CALL READ
CALL DELAY_L
```



```
CALL    DELAY_L
CALL    READ
CALL    DELAY_L
CALL    READ
CALL    DELAY_L
CALL    STOP
END
```

```
GO_F1TO3:      CALL    READ
```

```
CALL    DELAY_L
CALL    STOP
CALL    DELAY_L
CALL    STOP
CALL    ROTR
CALL    STOP
CALL    READ
CALL    DELAY_L
CALL    STOP
CALL    ROTR
CALL    STOP
END
```

```
GO_F1TO4:      CALL    READ
```

```
CALL    DELAY_L
CALL    STOP
CALL    ROTL
CALL    STOP
CALL    READ
CALL    DELAY_L
CALL    STOP
CALL    ROTL
```

```
CALL STOP
CALL DELAY_L
CALL STOP
CALL ROTL
CALL STOP
CALL DELAY_L
CALL READ
CALL DELAY_L
CALL STOP
END
GO_F2TO0: CALL READ
CALL DELAY_L
CALL STOP
CALL ROTR
CALL STOP
CALL READ
CALL DELAY_L
CALL STOP
CALL STOP
END
GO_F2TO1: CALL READ
CALL DELAY_L
CALL STOP
CALL ROTR
CALL STOP
CALL READ
CALL DELAY_L
CALL STOP
```

```
CALL      ROTR
CALL      STOP
CALL      READ
CALL      DELAY_L
CALL      STOP
CALL      ROTL
CALL      STOP
CALL      READ
CALL      DELAY_L
CALL      READ
CALL      DELAY_L
CALL      STOP
END
GO_F2TO3: CALL      READ
CALL      DELAY_L
CALL      STOP
CALL      READ
CALL      DELAY_L
CALL      STOP
CALL      ROTL
CALL      STOP
CALL      READ
CALL      DELAY_L
CALL      STOP
CALL      READ
CALL      DELAY_L
CALL      STOP
END
GO_F2TO4: CALL      READ
```

CALL DELAY_L

CALL STOP

CALL DELAY_L

CALL STOP

CALL ROTL

CALL STOP

CALL READ

CALL DELAY_L

CALL STOP

CALL READ

CALL DELAY_L

CALL STOP

END

GO_F3TO0:

CALL READ

CALL DELAY_L

CALL STOP

CALL ROTR

CALL STOP

CALL READ

CALL DELAY_L

CALL STOP

CALL DELAY_L

CALL STOP

CALL ROTL

CALL ROTR

CALL STOP

CALL READ

CALL READ

```
CALL      DELAY_L
CALL      READ
CALL      STOP
END
GO_F3TO1: CALL      READ
CALL      DELAY_L
CALL      STOP
CALL      STOP
CALL      ROTR
CALL      STOP
CALL      READ
CALL      DELAY_L
CALL      STOP
CALL      ROTL
CALL      STOP
END
GO_F3TO2: CALL      READ
CALL      DELAY_L
CALL      STOP
CALL      ROTR
CALL      STOP
CALL      READ
CALL      DELAY_L
CALL      STOP
CALL      READ
CALL      DELAY_L
CALL      READ
CALL      DELAY_L
```

```
CALL STOP
CALL ROTR
CALL STOP
END
```

```
GO_F3TO4: CALL READ
CALL DELAY_L
CALL STOP
CALL ROTR
CALL STOP
CALL READ
CALL DELAY_L
CALL STOP
CALL DELAY_L
CALL READ
CALL DELAY_L
CALL READ
CALL DELAY_L
CALL READ
CALL DELAY_L
CALL STOP
END
```

```
GO_F4TO0: CALL READ
CALL DELAY_L
CALL STOP
CALL DELAY_L
CALL STOP
CALL ROTL
CALL STOP
```

```
CALL READ
CALL DELAY_L
CALL STOP
CALL READ
CALL DELAY_L
CALL STOP
```

```
CALL ROTL
CALL STOP
CALL READ
CALL DELAY_L
CALL STOP
END
```

GO_F4TO1:

```
CALL READ
CALL DELAY_L
CALL STOP
CALL ROTL
CALL STOP
CALL READ
CALL DELAY_L
CALL STOP
CALL ROTR
CALL STOP
CALL READ
CALL DELAY_L
CALL STOP
CALL ROTR
CALL STOP
CALL READ
```

```
CALL    DELAY_L
CALL    STOP
CALL    READ
CALL    DELAY_L
CALL    STOP
END
```

```
GO_F4TO2:  CALL    READ
           CALL    DELAY_L
           CALL    STOP
           CALL    ROTL
           CALL    STOP
           CALL    READ
           CALL    DELAY_L
           CALL    STOP
           CALL    ROTR
           CALL    STOP
           CALL    READ
           CALL    DELAY_L
           CALL    STOP
           CALL    STOP
           CALL    READ
           CALL    DELAY_L
           CALL    STOP
           END
```

```
GO_F4TO3:  CALL    READ
           CALL    DELAY_L
           CALL    STOP
           CALL    ROTL
```



```
CALL STOP
CALL READ
CALL DELAY_L
CALL STOP
CALL ROTL
CALL STOP
CALL READ
CALL DELAY_L
CALL STOP
CALL ROTL
CALL STOP
CALL READ
CALL DELAY_L
CALL READ
CALL DELAY_L
CALL READ
CALL DELAY_L
CALL READ
CALL DELAY_L
CALL STOP
END
```

```
,*****
;
;DEFINE CONSTANT [STORE IN FLASH EEPROM PROGRAM MEMORY ]
,*****
;
;                                '0123456789ABCDEF'
TITLE_1:                        DB      '  AUTOMATIC  '
TITLE_2:                        DB      '  TRANSPORT  '
FROM:                            DB      '    FROM    '
SEND_TO:                        DB      '  SEND TO   '
```

ERROR:	DB	'	ERROR	'
FROM_0:	DB	'	FROM 0	'
FROM_1:	DB	'	FROM 1	'
FROM_2:	DB	'	FROM 2	'
FROM_3:	DB	'	FROM 3	'
FROM_4:	DB	'	FROM 4	'
FROM_0TO1:	DB	'	FROM 0 SEND TO 1	'
FROM_0TO2:	DB	'	FROM 0 SEND TO 2	'
FROM_0TO3:	DB	'	FROM 0 SEND TO 3	'
FROM_0TO4:	DB	'	FROM 0 SEND TO 4	'
FROM_1TO0:	DB	'	FROM 1 SEND TO 0	'
FROM_1TO2:	DB	'	FROM 1 SEND TO 2	'
FROM_1TO3:	DB	'	FROM 1 SEND TO 3	'
FROM_1TO4:	DB	'	FROM 1 SEND TO 4	'
FROM_2TO0:	DB	'	FROM 2 SEND TO 0	'
FROM_2TO1:	DB	'	FROM 2 SEND TO 1	'
FROM_2TO3:	DB	'	FROM 2 SEND TO 3	'
FROM_2TO4:	DB	'	FROM 2 SEND TO 4	'
FROM_3TO0:	DB	'	FROM 3 SEND TO 0	'
FROM_3TO1:	DB	'	FROM 3 SEND TO 1	'
FROM_3TO2:	DB	'	FROM 3 SEND TO 2	'
FROM_3TO4:	DB	'	FROM 3 SEND TO 4	'
FROM_4TO0:	DB	'	FROM 4 SEND TO 0	'
FROM_4TO1:	DB	'	FROM 4 SEND TO 1	'
FROM_4TO2:	DB	'	FROM 4 SEND TO 2	'
FROM_4TO3:	DB	'	FROM 4 SEND TO 3	'

END

ในการแสดงตำแหน่งของรถขนส่งอัตโนมัติผ่านทางหน้าจอคอมพิวเตอร์โดยใช้โปรแกรม
วิชวลเบสิก (Visual Basic) ซึ่งเขียนโค้ด (Code) ได้ดังนี้

```
Dim recive
```

```
Const Cred = &HFF
```

```
Const Cwhite = &HFFFFFF
```

```
Private Sub Command1_Click()
```

```
If MSComm1.PortOpen = False Then
```

```
    Shape1.Visible = True
```

```
    MSComm1.CommPort = 1 ' Use COM1.
```

```
    MSComm1.Settings = "9600,N,8,1" ' 9600 baud, no parity, 8 data, and 1 stop bit.
```

```
    MSComm1.PortOpen = True
```

```
End If
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
If MSComm1.PortOpen = True Then
```

```
    Buffer = Text1
```

```
    MSComm1.Output = Buffer
```

```
End If
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
    Shape1.Visible = False
```

```
    MSComm1.PortOpen = False
```

```
End Sub
```

```
Private Sub Command5_Click()
```

```
    recive = ""
```

```
    Text2 = recive
```

```
End Sub
```

```
Private Sub Command6_Click()
```

```
Sta (Text1)
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Shape1.Visible = False
```

```
Command1.Caption = "Open port"
```

```
Command2.Caption = "Send Data"
```

```
Command3.Caption = "Close port"
```

```
Command5.Caption = "Clear buffer"
```

```
Call Clear
```

```
End Sub
```

```
Private Sub MSComm1_OnComm()
```

```
Dim a
```

```
a = MSComm1.Input
```

```
'recive = Val("&h" + a)
```

```
Text1 = recive
```

```
Call Sta(recive)
```

```
End Sub
```

```
Sub Sta(id)
```

```
If id = 256 Then
```

```
Call Clear
```

```
ShpA.BackColor = Cred
```

```
Elseif id = "512" Then
```

```
Call Clear
```

```
ShpB.BackColor = Cred
```

```
ElseIf id = "1024" Then
Call Clear
ShpC.BackColor = Cred
ElseIf id = "2048" Then
Call Clear
ShpD.BackColor = Cred
ElseIf id = "4096" Then
Call Clear
ShpE.BackColor = Cred
ElseIf id = "8192" Then
Call Clear
ShpF.BackColor = Cred
ElseIf id = "16384" Then
Call Clear
ShpG.BackColor = Cred
ElseIf id = "32768" Then
Call Clear
ShpH.BackColor = Cred
ElseIf id = "1" Then
Call Clear
ShpI.BackColor = Cred
ElseIf id = "2" Then
Call Clear
ShpJ.BackColor = Cred
ElseIf id = "4" Then
Call Clear
ShpK.BackColor = Cred
End If

End Sub
Sub Clear()
```

ShpA.BackColor = Cwhite

ShpB.BackColor = Cwhite

ShpC.BackColor = Cwhite

ShpD.BackColor = Cwhite

ShpE.BackColor = Cwhite

ShpF.BackColor = Cwhite

ShpG.BackColor = Cwhite

ShpH.BackColor = Cwhite

ShpI.BackColor = Cwhite

ShpJ.BackColor = Cwhite

ShpK.BackColor = Cwhite

End Sub