สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หุ่นยนต์จดจำเส้นทาง

Route Recognition Robot

โดย

นายวีระศักดิ์ เปรมนิวัติ

นายเอกชัย สุนันท์ชัย

เลขหมู่.....
เลขทะเบียน 46271
วัน, เดือน, ปี 21 มี.ค. 2546

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์จดจำเส้นทาง

Route Recognition Robot

โดย

นายวีระศักดิ์ เปรมนิวัติ เลขประจำตัว 42015239

นายเอกชัย สุนันท์ชัย เลขประจำตัว 42015251

อาจารย์ที่ปรึกษา

ดร.ยุทธนา คิดใจเดียว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

หุ่นยนต์จดจำเส้นทาง
ผู้จัดทำ

1.นายวีระศักดิ์ เปรมนิวัติ
2.นายเอกชัย สุนันท์ชัย

โครงงานนี้ได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(ดร.ยุทธนา คิดใจเดียว)
อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์จดจำเส้นทาง

นายวีระศักดิ์ เปรมนิวัติ
นายเอกชัย สุนันท์ชัย
ดร.ยุทธนา คิดใจเดียว
ภาคเรียนที่ 2 ปีการศึกษา 2544

บทคัดย่อ

โดยปกติการแก้ปัญหาต่างๆด้วยโปรแกรมคอมพิวเตอร์นั้นจะเริ่มต้นที่การพิจารณาอินพุต แล้วออกแบบขบวนการที่จะดำเนินการกับอินพุตเหล่านั้นเพื่อจะได้สร้างเอาต์พุตได้ตามต้องการ วิธีการพัฒนาโปรแกรมคอมพิวเตอร์แบบคลาสสิกนี้มีข้อจำกัดคือไม่สามารถใช้ได้กับปัญหาซึ่งขบวนการมีความซับซ้อนอย่างมากจนเราไม่สามารถสรุปเป็นขั้นตอนที่แน่นอนได้ ระบบโครงข่ายแบบฉลาดเป็นรูปแบบการแก้ปัญหาอีกแบบหนึ่งซึ่งเหมาะกับปัญหาดังกล่าว ระบบโครงข่ายแบบฉลาดจะสร้างขบวนการขึ้นเองจากความรู้ซึ่งได้จากการเรียนรู้จากตัวอย่าง ถ้าตัวอย่างที่นำมาสอนมีจำนวนมากพอและมีวิธีการสอนดีพอ ขบวนการที่ถูกสร้างขึ้นจะมีความถูกต้องเพียงพอสำหรับนำไปใช้งานจริงได้

สำหรับโครงงานนี้เป็นการจำลองลักษณะการทำงานของระบบโครงข่ายแบบฉลาดโดยการนำมาประยุกต์ใช้กับการจำเส้นทางการเคลื่อนที่ของหุ่นยนต์โดยให้หุ่นยนต์เดินทางตามเส้นทางที่เรากำหนดไว้ และหลังจากนั้นหุ่นยนต์ก็จะเคลื่อนที่ไปเป็นเส้นทางเดิมได้เอง โดยจะใช้หน่วยประมวลผลกลางคือ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Route Recognition Robot

Mr.Verasak Pramenivut

Mr.Akechai Suunanchai

Prof.Yuttana Kidjaidaew Adviser

2001

Abstract

In general computer solving problem, programmer will firstly consider an input and step forward along the designed algorithm to produce an output. Classical programs have a lot of limit conditions using with a complicated process. So an artificial intelligence network is one way to solve this problem which can manage things through by examples.

This project present the recognize of route by robot. Robot can move to route and recall route by itsalf.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญรูปภาพ	V
บทที่ 1บทนำ	1
บทที่ 2 ทฤษฎีนิวรอลเน็ตเวิร์ค	2
2.1 Weightless Neural Network	2
2.2การประยุกต์ใช้ระบบโครงข่ายประสาทเทียม	5
บทที่ 3 ทฤษฎีไมโครคอนโทรลเลอร์ตระกูล 8051	6
3.1 คุณลักษณะพื้นฐานของ 8051	6
3.2 ลักษณะการจัดขาของ MCS-51	7
3.3 พอร์ตอินพุตและพอร์ตเอาท์พุต	9
3.4 หน่วยความจำโปรแกรมของ 8051	11
3.5 หน่วยความจำข้อมูลของ 8051	12
3.6 รีจิสเตอร์หน้าที่พิเศษ	13
บทที่ 4 การออกแบบอัลกอลิทึม	15
4.1 การออกแบบอินพุทและเอาท์พุท ของระบบแยกแยะความแตกต่าง	15
4.2 การสร้าง Lookup-table ของระบบ แยกแยะความแตกต่างในสถาวะรีคอกไนซ์(Recognize)	16
4.3 การค้นหาเส้นทาง	19
4.3.1 การตัดสินใจเลือกเส้นทาง	19
บทที่ 5 หลักการออกแบบวงจร	25
5.1 โครงสร้างฮาร์ดแวร์(Hardware)	25
5.1.1 วงจรเซ็นเซอร์	25
5.1.2 วงจรขับมอเตอร์	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

5.1.3 ส่วนของการควบคุมการทำงานโดยใช้ไมโครคอนโทรลเลอร์ MCS-51 27

บทที่ 6 การทดลองและบทสรุป 30

6.1 การทดลอง 30

6.2 สาเหตุของปัญหา 31

6.3 แนวทางการแก้ไขปัญหาและพัฒนา 31

6.4 สรุปผลการทดลอง 31

ภาคผนวก

กิตติกรรมประกาศ

เอกสารอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 บล็อกไดอะแกรมพื้นฐานของนิวรอลเน็ทเวิร์ค แบบที่ใช้ RAMs 3
รูปที่ 3.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ตระกูล 8051 7
รูปที่ 3.2 แสดงการจัดขาของไมโครคอนโทรลเลอร์ MCS-51 7
ตารางที่ 3.1 แสดงหน้าที่พิเศษของแต่ละขาของพอร์ต 3 ของไมโครคอนโทรลเลอร์ 8
รูปที่ 3.3a เป็นการใช้ไมโครคอนโทรลเลอร์เป็นเอาท์พุตพอร์ต 9
รูปที่3.3b เป็นการใช้ไมโครคอนโทรลเลอร์เป็นอินพุตพอร์ต 10
รูปที่3.4 การจัดพื้นที่ของหน่วยความจำโปรแกรมของ ไมโครคอนโทรลเลอร์ 8051 11
รูปที่ 3.5 ลักษณะตำแหน่งที่ตั้งของหน่วยความจำข้อมูล 12
รูปที่ 3.6 แสดงถึงการจัดพื้นที่ของหน่วยความจำข้อมูลภายในของ 8051 13
รูปที่ 4.1 กลุ่มข้อมูลอินพุทของระบบระบบแยกแยะความแตกต่าง 15
รูปที่ 4.2 ข้อมูลในแต่ละเหตุการณ์ 15
รูปที่ 4.3 แสดงการเลื่อนของข้อมูลเมื่อช่วงเวลาเปลี่ยนไป 16
รูปที่ 4.4 แสดงแบบจำลองระบบแยกแยะความแตกต่าง ของการเก็บข้อมูลไว้ในLookup-table ของ ระบบแยกแยะความแตกต่างแต่ละตัว 17
รูปที่ 4.5 แสดงข้อมูลที่ถูกกำหนดให้เป็นอินพุทและเอาท์พุท ให้ระบบแยกแยะความแตกต่างแต่ละตัว 18
รูปที่4.6 แสดงตารางการกำหนดอินพุทและเอาท์พุทให้ระบบ แยกแยะความแตกต่างแต่ละตัว 18
รูปที่ 4.7 แผนผังของช่วงการจำเส้นทาง 20
รูปที่ 4.8 แผนผังของการรีคอลวิธีที่ 1 21
รูปที่ 4.9 แผนผังของการรีคอลวิธีที่ 2 22
รูปที่ 4.10 แผนผังของการรีคอลวิธีที่ 3 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.11 แผนผังของการรีคอลวิธีที่ 3 24
รูปที่ 5.1 วงจรเซ็นเซอร์ 26
รูปที่ 5.2 วงจรขับมอเตอร์ 26
รูปที่ 5.3 วงจรระบบโครงข่ายแบบฉลาด 28
รูปที่ 5.4 บล็อคไดอะแกรมระบบโครงข่ายแบบฉลาด 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

โดยปกติการแก้ปัญหาต่างๆด้วยโปรแกรมคอมพิวเตอร์นั้นจะเริ่มต้นที่การพิจารณาอินพุต แล้วออกแบบขบวนการที่จะดำเนินการกับอินพุตเหล่านั้นเพื่อจะได้สร้างเอาต์พุตได้ตามต้องการ วิธีการพัฒนาโปรแกรมคอมพิวเตอร์แบบดั้งเดิมนี้มีข้อจำกัดคือไม่สามารถใช้ได้กับปัญหาซึ่งขบวนการมีความซับซ้อนอย่างมากจนเราไม่สามารถสรุปเป็นขั้นตอนที่แน่นอนได้ ระบบโครงข่ายแบบฉลาดเป็นรูปแบบการแก้ปัญหาอีกแบบหนึ่งซึ่งเหมาะกับปัญหาดังกล่าว ระบบโครงข่ายแบบฉลาดจะสร้างขบวนการขึ้นเองจากความรู้ซึ่งได้จากการเรียนรู้จากตัวอย่าง ถ้าตัวอย่างที่นำมาสอนให้แก่ระบบโครงข่ายแบบฉลาดมีจำนวนมากพอและมีวิธีการสอนดีพอ ขบวนการที่ถูกสร้างขึ้นจะมีความถูกต้องเพียงพอสำหรับนำไปใช้งานจริงได้

วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาโครงสร้างของระบบโครงข่ายแบบฉลาด
2. ออกแบบอัลกอลิทึมเพื่อให้มีประสิทธิภาพมากที่สุด
3. ศึกษาวิธีการเขียนโปรแกรมเพื่อที่จะนำไปเขียนโปรแกรมต่างในอนาคต
4. เพื่อที่จะนำระบบโครงข่ายแบบฉลาดไปประยุกต์ใช้งานในด้านต่างๆ

รายละเอียดของโครงงาน

สำหรับโครงงานนี้เป็นการจำลองลักษณะการทำงานของระบบโครงข่ายแบบฉลาดโดยการนำมาประยุกต์ใช้กับการจำเส้นทางการเคลื่อนที่ โดยจำลองการเรียนรู้ของระบบโครงข่ายแบบฉลาดโดยให้หุ่นยนต์เดินทางตามเส้นทางที่เรากำหนดไว้ และหลังจากนั้นก็จะเคลื่อนที่ไปเป็นเส้นทางเดิมที่ได้เรียนรู้มาได้เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ทฤษฎีนิวรอลเน็ตเวิร์ค

นิวรอลเน็ตเวิร์ค คือระบบคอมพิวเตอร์ที่มีโครงสร้างและการทำงานในรูปแบบการเรียนรู้แบบ content-addressable memory คุณลักษณะที่สำคัญที่นำมาประยุกต์ใช้เป็นพื้นฐานของระบบโครงข่ายแบบฉลาดคือสถาปัตยกรรมแบบกระจาย(Distributed) และแบบรวม(Local represent) ของเซลล์ต่างๆ และความสามารถในการเรียนรู้และในการปรับตัวตามข้อมูลที่ได้รับจากสภาพแวดล้อมภายนอก

2.1 Weightless Neural Network

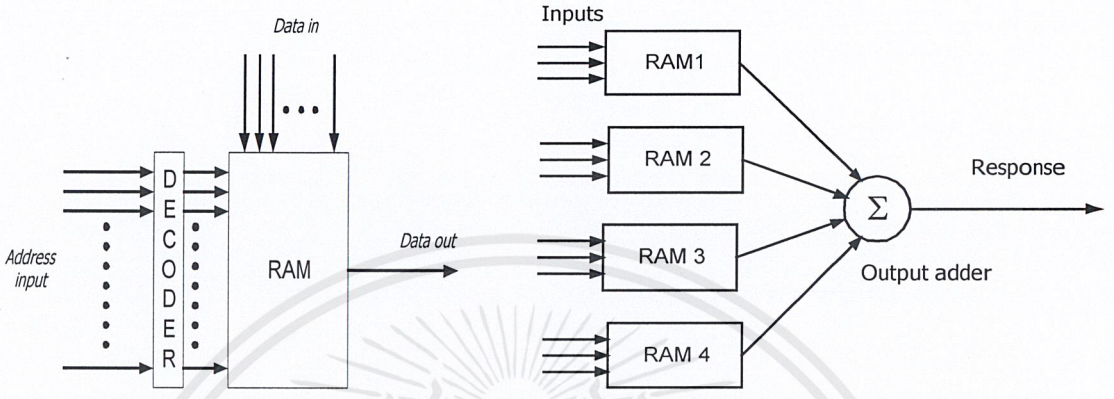
Weightless models คือ RAM ที่เป็นพื้นที่สำหรับเก็บข้อมูลการกระตุ้นของอินพุทและผลตอบสนองต่อความสัมพันธ์ในLook-up table ไปในรูปแบบของปฏิกริยาทางตรรกศาสตร์ของนิวรอล ต้นแบบความคิดนี้ถูกเสนอโดย Bledsoe และ Browning(1959) ซึ่งรู้จักกันในชื่อของกระบวนการจำแนกข้อมูลแบบ n-tuple(n-tuple recognition process) tuple แต่ละยูนิตจะรับข้อมูลเพียงส่วนเล็กๆของข้อมูลทั้งหมดและตอบสนองโดยผลที่ได้ของแต่ละยูนิตเป็นอิสระต่อกัน การจำแนกข้อมูลใน n-tuple machine สามารถทำได้โดยเปรียบเทียบลักษณะที่คล้ายกันระหว่างรูปแบบของข้อมูลที่ป้อนเข้าไปโดยให้ลักษณะที่คล้ายคลึงกันออกมาในรูปของคะแนน ซึ่งข้อได้เปรียบของเทคนิคนี้คือข้อมูลหลายตัวสามารถมีผลตอบสนองร่วมกันใน tuple หนึ่งยูนิตและสามารถแยกรูปแบบของข้อมูลแบบไม่เป็นเชิงเส้น(non-linearly) ออกมาได้

ระหว่างทศวรรษที่ 70 การพัฒนาเทคโนโลยีของวงจรรวมหรือ (Integrated Circuit Technology) สามารถสร้าง Random Access Memories (RAMs) ได้ซึ่งเป็นการนำไปสู่การค้นพบนิวรอลเน็ทเวิร์คที่ใช้หน่วยความจำแบบ RAMs เป็นที่เก็บข้อมูลของ n-tuple machines โดย Aleksander และ Stonham (1979) หลังจากนั้นหลักการเดียวกันนี้ถูกนำไปใช้ในการออกแบบ WISARD (Aleksander,Thomas and Bowden 1984, Aleksander 1984)

เมื่อพิจารณาการใช้นิวรอลเน็ทเวิร์คแบบที่ใช้RAMแล้วพบว่ามีข้อดีในการเรียนรู้ algorithm เนื่องจากแต่ละโหนดอิสระต่อกันระหว่างการอ้างอิงข้อมูลจาก Look-up table เว้นแต่ในส่วนของ weighted models ในส่วนที่สองซึ่งน้ำหนักของข้อมูลต้องขึ้นกับการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนแปลงในส่วนแรก ส่วนข้อดีอื่นๆนั้นคือฮาร์ดแวร์ซึ่งประกอบด้วย RAMs มีราคาถูกและสามารถหาได้ง่ายโดยบล็อกไดอะแกรมพื้นฐานของนิวรอลเน็ทเวิร์คแบบที่ใช้ RAMs นี้สามารถแสดงได้ดังรูปที่ 2.1



รูปที่ 2.1(a) A RAM-based neural (b) 4 RAMs discriminators with 3 input each

โมเดลของนิวรอลแบบไร้น้ำหนัก(weightless neural model หรือ WISARD)คือระบบที่มีการป้อนไปข้างหน้าแบบชั้นเดียวที่ประกอบไปด้วยเซ็ตสัญลักษณ์การแบ่งแยกคลาสของdiscriminator รูปแบบของข้อมูลตัวแบ่งแยกทั้งหมดจะเริ่มต้นที่เซ็ตของศูนย์และค่อยๆเปลี่ยนไปตามการป้อนข้อมูล(training) ระหว่างการป้อนข้อมูลรูปแบบของข้อมูลแต่ละอันก็จะถูกเก็บอยู่แค่ในRAMของการตอบสนองการแบ่งแยกหลังจากการป้อนข้อมูลเสร็จสิ้นก็จะได้ผลการแบ่งแยกของแต่ละคลาส การจำแนกข้อมูล(recognition) ก็จะเริ่มโดยป้อนรูปแบบของข้อมูลที่ไม่ทราบค่าให้กับdiscriminatorsทุกๆตัวเพื่อหาผลตอบสนองที่มีค่ามากที่สุดโดยหลักการทั่วๆไปของการคิดคือไม่นำมาคิดแค่เพียงแต่ละโหนดแต่คิดทั้งระดับของเน็ทเวิร์ค

เนื่องจากการทำงานเป็นเส้นตรงของ WISARD จะมีความคลุมเครือระหว่างสถานะ 0 ที่เกิดขึ้นระหว่างการเก็บรูปแบบของข้อมูลและสถานะ 0 ที่จุดเริ่มต้นก่อนจะเกิดการป้อนข้อมูลหรือtraining เพื่อแก้ปัญหาข้อนี้ Ken และ Aleksander (1988) ได้เสนอการใช้ Probabilitic Logic Node (PLN) ซึ่งแต่ละโหนดจะยังคงมีเอาท์พุทเป็นเลขฐานสองแต่จะมีสถานะการทำงานเป็น 3 ระดับ โดยสถานะที่เพิ่มขึ้นมาก็คือ u-state ซึ่งจะแทนตรงส่วนที่ค่าเริ่มต้นหรือในช่วงที่ข้อมูลยังไม่ถูกเก็บ ซึ่งโอกาสที่เอาท์พุทของโหนดจะเป็น 1 และ 0 จะมีค่า 50% ในrecalling mode ในระหว่างการ Training แต่ละ pattern จะถูกเก็บไว้ใน RAM ของ นิวรอล ที่ตรงกันเท่านั้น หลังจาก Training ในแต่ละ Class เสร็จเรียบร้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้ว การรู้จำสมบูรณ์แบบก็จะเกิดขึ้นจากการที่เราป้อน unknows pattern ให้กับ ระบบโครงข่ายแบบฉลาด ทุกตัวแล้วทำการหาคำตอบที่แน่นอน

การพัฒนาของ PLNs ที่เป็นที่รู้จักกันในนามของ Goal Seeking Neural (GNS) ซึ่งจะต่างจากPLNs คือจะมี u-state เป็นแบบการแพร่ผ่าน Network แทนที่จะเป็นการบังคับ node เพื่อที่จะได้เอาท์พุทออกมาระหว่าง 0 และ 1 ซึ่ง Multi-addressing ใน GSNs มีจุดประสงค์เพื่อลดจำนวนของ u-state location และเป็นการหลีกเลี่ยงความลังเลระหว่าง retrival phase เนื่องจาก u-state ที่เข้ามาเป็นจำนวนมาก

Version อื่นๆ ของ RAM-base neurons ที่ประกอบด้วยค่าจริงที่ต่อเนื่องที่มีชื่อว่า The multi-value probabilitic RAMs(pRAKs) ซึ่งแทนที่จะมีเพียงค่า binary เก็บไว้ใน look-up table model นี้ยังอนุญาตให้ input อยู่ในรูปแบบของค่าจริงที่มีความต่อเนื่อง ความน่าจะเป็นของ 1 ที่ output จะคำนวนได้จากการถ่วงน้ำหนักผลรวมของเวกเตอร์ทางด้านอินพุทต่อเนื่อง

Version ต่อมาเรียกว่า Cascaded Single Weightless Neural Network (CSWNN) โมเดลนี้จะประกอบด้วยกลุ่มของ RAM Neurons ซึ่งแต่ละกลุ่มจะมี feedback ต่ออยู่ภายในตัวของมันเอง และเพื่อต่อโดยตรงกับ Network กลุ่มต่อไป (Cascade network)เป็นกลไกเฉพาะเพื่อที่จะทำหน้าที่เกี่ยวกับปัญหาเรียนรู้ด้านภาษา

เนื่องจากใช้แค่เพียง Location ซึ่งหามาจากที่เรา trained patterns เป็นการพัฒนาระหว่างการ training PLNs อาจเกิดความเสียหายจากการ delay เมื่อ training set มีค่าน้อยเกินไป (เช่นมีประมาณ 2-3 ชุด)ความเสียหายอื่นๆ ที่เป็น generalization เกิดขึ้นในระดับชั้น Network เทคนิคของการ Generalization ที่ไม่สนใจอินพุทที่ node level เรียกว่า spreading process ซึ่งเป็นขบวนการที่ต่อจากช่วง learning phase กระบวนการทำงานโดยการกำหนดค่า u-state location จากค่าที่ใกล้เคียงที่สุดใน Hamming distance

ถ้ามีการขัดแย้งระหว่างค่าที่ใกล้เคียงที่สุด (เลือกค่าใดค่าหนึ่งไม่ได้) Location ของ u-state จะไม่มีการเปลี่ยนแปลงค่า generalization ของ network ขึ้นอยู่กับระยะรัศมีของ Spreading regions ที่แต่ละ node ถ้าค่ารัศมีเท่ากับจำนวนทั้งหมดของ binary output หมายความว่า network มีการกระจายเต็มซึ่ง network ที่มีคุณสมบัติดังนี้เราจะเรียกว่า Generalizing RAMs (GRAMs) ขนาดของ Generalizing ใน GRAMs จะถูกกำหนดโดยจำนวนของ patterns ในการ training เพราะว่า spreading region จะมีขนาดเล็กลงเมื่อจำนวน pattern มา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การประยุกต์ใช้ระบบโครงข่ายประสาทเทียม(Artificial Neural Network Application)

ในปัจจุบันได้มีการนำนิวรอนเน็ตเวิร์คมาประยุกต์ใช้งานหลายด้าน ตัวอย่างกลุ่มงานที่ได้รับความนิยมมีดังนี้

Recognition: การนำนิวรอนเน็ตเวิร์คมาประยุกต์ใช้กับการจดจำได้รับการนิยมมากที่สุดในปัจจุบัน เนื่องจากนิวรอนเน็ตเวิร์คมีความสามารถที่จะจดจำรูปแบบที่มีความซับซ้อนมากดี ตัวอย่างของเน็ตเวิร์คที่เหมาะกับงานในด้านนี้คือ มัลติเลเยอร์เพอเซพตรอน

Clustering: หลักการของงานด้านนี้คือ อินพุทเน็ตเวิร์คที่มีคุณสมบัติหรือลักษณะใกล้เคียงกันจะถูกจัดให้อยู่ในกลุ่มเดียงกัน เน็ตเวิร์คที่เหมาะสมกับงานด้านนี้คือ อันซุปเปอร์ไวส์นิวรอนเน็ตเวิร์ค ตัวอย่างเช่น ART Network และ Kohonen's Self-organizing feature map

Classification: ในขั้นตอนการปรับสอน จำนวนคลาส (Class) และคุณสมบัติของแต่ละคลาสจะถูกป้อนเข้าสู่เน็ตเวิร์ค อินพุทของเน็ตเวิร์คจะถูกส่งเข้าคลาสที่เหมาะสม เน็ตเวิร์คที่เหมาะสมกับงานด้านนี้คือ มัลติเลเยอร์เพอเซพตรอน และ Linear Vector Quantization(LVQ)เน็ตเวิร์ค

Forecasting: อัลกอริทึมการเรียนรู้ที่ใช้กับงานด้านนี้คือ ซุปเปอร์ไวส์อัลกอริทึม เน็ตเวิร์คจะถูกปรับสอนด้วยชุดข้อมูลในอดีต ซึ่งจะประกอบด้วยอินพุทและเอาท์พุทเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3
ทฤษฎีไมโครคอนโทรลเลอร์ตระกูล 8051

3.1 คุณลักษณะพื้นฐานของ 8051

คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ประกอบไปด้วย

1.หน่วยประมวลผลกลางขนาด 8 บิต

2.หน่วยประมวลผลสำหรับข้อมูลแบบบิต (Boolean Processor)

3.ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม 64 กิโลไบต์

4.ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์

5.หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ แบบ EPROM (เบอร์ 8751) หรือแบบ ROM (เบอร์ 8051)

6.หน่วยความจำแบบ RAM ภายในจำนวน 128 ไบต์

7.พอร์ตอินพุตเอาต์พุตแบบขนานจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ

8.วงจรนับ/จับเวลาขนาด 16 บิตจำนวนสองวงจร

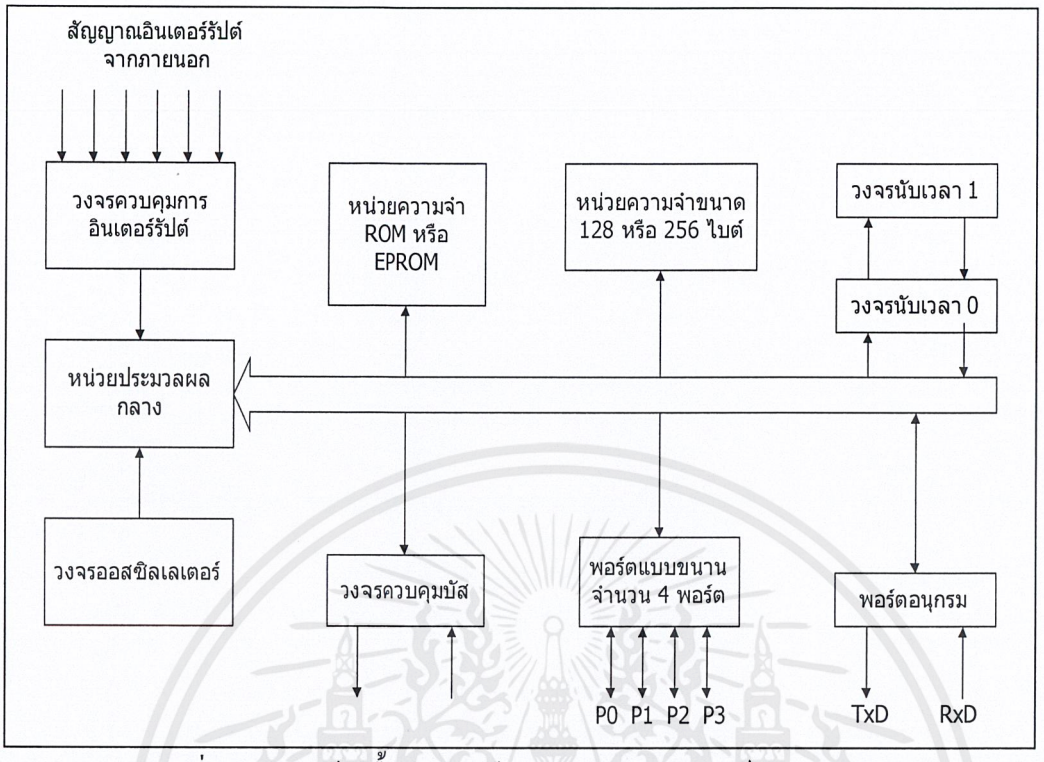
9.วงจรสื่อสารแบบอนุกรมแบบฟูลดูเพล็กซ์ (Full Duplex)

10.วงจรควบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมกำหนดความสำคัญได้สองระดับ

11.มีวงจรออสซิลเลเตอร์ภายใน

โดยโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ตระกูล8051 สามารถแสดงได้ดังรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ตระกูล 8051

3.2 ลักษณะการจัดขาของ MCS-51

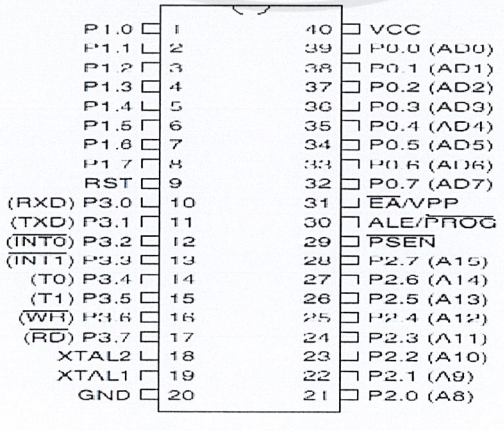
จากรูปที่ 3.2 เป็นการจัดขาตามตามลักษณะภายนอกของ MCS-51 ซึ่งจะแบ่งการจัดขาของ MCS-51 จะมีอยู่ 4 กลุ่มคือ

1.กลุ่มขาแหล่งจ่ายไฟเลี้ยงและสัญญาณนาฬิกา

2.กลุ่มขาสำหรับอ้างแอดเดรสและรับส่งข้อมูล

3.กลุ่มขาที่ใช้ในการควบคุม

4.กลุ่มขาพอร์ตใช้งานแบบขนานและอนุกรม



รูปที่ 3.2 แสดงการจัดขาของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยขาที่สำคัญของไมโครคอนโทรลเลอร์ MCS-51มีดังนี้

1. ขา Vcc เป็นขารับแรงดันไฟตรง +5 โวลท์

2. ขา GND เป็นขากราวด์

3. พอร์ต 0 (Port 0) มี 8 บิต ได้แก่บิต P0.0-P0.7 เป็นพอร์ตเอาท์พุตแบบ 2 ทิศทาง สำหรับใช้งานทั่วไปโดยถ้าใช้งานเป็นอินพุตพอร์ต ต้องทำการเซ็ตค่า 1 ไปยังพอร์ตทั้งพอร์ต และถ้าต้องการใช้งานเป็นพอร์ตอินพุตในระดับบิตก็สามารถทำได้โดยการเซ็ตค่า 1 ไปยังแต่ละบิตที่ต้องการใช้งานเป็นพอร์ตอินพุตในระดับบิต นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตเอาท์พุตแล้วยังใช้งานในการติดต่อกับหน่วยความจำภายนอก(EPROM,ROM)ได้อีกด้วยโดยการกำหนดแอดเดรสไบต์ต่ำ (A0-A7) ซึ่งจะใช้งานเป็นมัลติเพล็กสำหรับการส่งข้อมูลขนาด 8 บิต (D0-D7)

1. พอร์ต 1 (Port 1) มี 8 บิต ได้แก่บิต P1.0-P1.7 เป็นพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทาง สำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตหรืออินพุตในระดับบิต สามารถกระทำได้โดยวิธีเช่นเดียวกันกับพอร์ต 0 ข้างต้น

2. พอร์ต 2 (Port 2) มี 8 บิต ได้แก่บิตP2.0-P2.7 เป็นพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทาง สำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตหรืออินพุตในระดับบิต สามารถกระทำได้โดยวิธีเช่นเดียวกันกับพอร์ต 0 ข้างต้น นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตเอาท์พุตแล้วยังใช้งานในการติดต่อกับหน่วยความจำภายนอก(EPROM,ROM)ได้อีกด้วยโดยการอ้างอิงตำแหน่งแอดเดรสไบต์สูง (A8-A15)

3. พอร์ต 3(Port 3) มี 8 บิต ได้แก่บิต P3.0-P3.7 เป็นพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทาง สำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตหรืออินพุตในระดับบิต สามารถกระทำได้โดยวิธีเช่นเดียวกันกับพอร์ต 0 ข้างต้น นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตพิเศษต่างๆดังตารางที่ 3.1

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

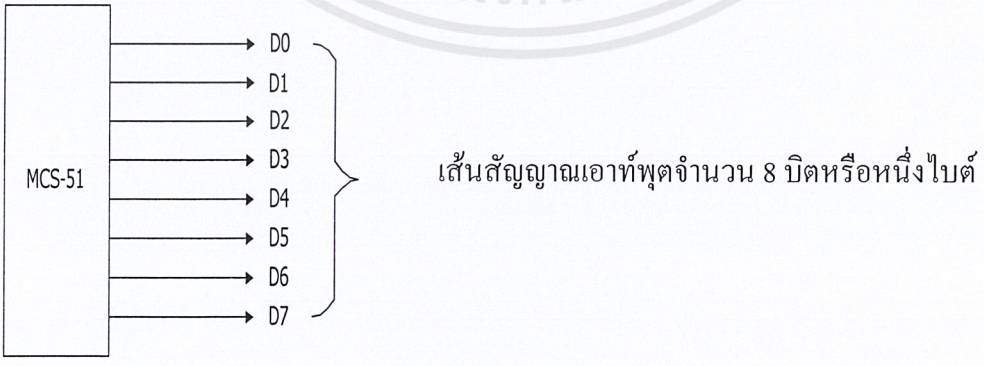
ตารางที่ 3.1 แสดงหน้าที่พิเศษของแต่ละขาของพอร์ต 3 ของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ขารีเซต (RES) ใช้สำหรับการรีเซตการทำงานของไมโครคอนโทรลเลอร์โดยการรีเซตต้องให้ขา นี้เป็น HIGH อย่างน้อยนาน 2 Machine Cycle ในขณะที่ออสซิลเลเตอร์ยังทำงานอยู่
2. ขา *ALE/* $\overline{PROG}$ เป็นขาสัญญาณเพื่อทำหน้าที่ควบคุมการแล็ช (Latch) ค่าตำแหน่งแอดเดรสไบต์ต่ำ (Address Latch Enable) เมื่อต้องการติดต่อกับหน่วยความจำภายนอกนอกจากนี้ขานี้ยังทำหน้าเป็นอินพุตรับพัลส์ในการโปรแกรม (Program Pulse Input) ในส่วนของหน่วยความจำ EPROM สำหรับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ที่มีหน่วยความจำโปรแกรมภายในเป็น EPROM
3. ขา $\overline{PSEN}$ (Program Store Enable) ทำหน้าที่เป็นสัญญาณสโตบเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอกเมื่อไมโครคอนโทรลเลอร์ประมวลคำสั่งจากหน่วยความจำภายนอก เมื่อไมโครคอนโทรลเลอร์ประมวลผลจากหน่วยความจำภายนอก ขานี้จะส่งสัญญาณสโตบจำนวน 2 ครั้งในแต่ละ Machine Cycle
4. ขา $\overline{EA/Vcc}$ (External Access Enable/Vcc) เป็นขาสำหรับเลือกใช้หน่วยความจำโปรแกรมจากภายในหรือจากภายนอก
5. ขา XTAL1 และขา XTAL2 เป็นขาใช้งานของวงจรอินเวอร์ติ้งออสซิลเลเตอร์แอมปลิไฟเออร์ (Inverting Oscillator Amplifier) สำหรับใช้ต่อร่วมกับคริสตอลภายนอก

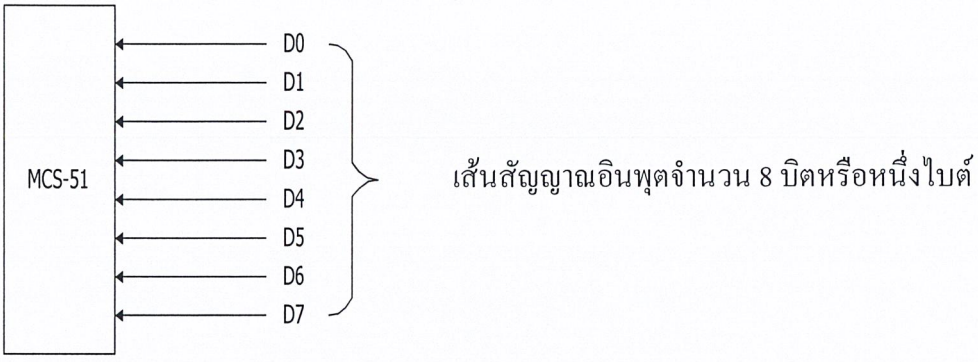
3.3 พอร์ตอินพุตและพอร์ตเอาท์พุต

พอร์ตคือแอดเดรสหนึ่งที่ได้รับการกำหนดไว้เพื่อการโอนย้ายข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก การกำหนดประเภทของการติดต่อขึ้นอยู่กับทิศทางการไหลของข้อมูลเมื่อพิจารณาจากไมโครคอนโทรลเลอร์เป็นหลัก จากรูปที่ 3.3a เป็นการใช้ไมโครคอนโทรลเลอร์เป็นเอาท์พุตพอร์ต และจากรูป 3.3b เป็นการใช้ไมโครคอนโทรลเลอร์เป็นอินพุตพอร์ต



รูปที่ 3.3a เป็นการใช้ไมโครคอนโทรลเลอร์เป็นเอาท์พุตพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.3b เป็นการใช้ไมโครคอนโทรลเลอร์เป็นอินพุตพอร์ต

3.3.1 การใช้งานพอร์ตเป็นการอินพุต

การใช้งานพอร์ตเป็นการอินพุตข้อมูลจะต้องเริ่มต้นด้วยการส่งข้อมูลที่มีค่าเป็น 1 ออกมาทางบิตของพอร์ตนั้นก่อนเป็นอันดับแรก เพื่อหยุดการทำงานของทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุตของบิตนั้น ทำให้ขาสัญญาณของบิตถูกต่อเข้ากับตัวต้านทานซึ่งทำหน้าที่ Pull-up ภายในซึ่งมีผลทำให้บิตนั้นของพอร์ต 1,2,3 เป็นสภาวะของลอจิกสูง ตัวต้านทานนี้มีค่าประมาณ 50 K โอห์ม ซึ่งเป็นค่าที่สูงมาก และทำให้อุปกรณ์ภายนอกสามารถขับสัญญาณของพอร์ตเหล่านี้เป็นลอจิกต่ำได้ง่าย สำหรับบิตของพอร์ต 0 นั้นแม้ว่าจะมีหลักการทำงานที่คล้ายคลึงกันกับบิตของพอร์ตอื่นๆ แต่เนื่องจากไม่มีตัวต้านซึ่งทำหน้าที่ Pull-up ภายในไว้ ทำให้เมื่อทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุตนั้นหยุดการทำงาน ก็จะเป็นผลให้สัญญาณนี้อยู่ในสภาวะอิมพีแดนซ์สูงแทน

3.3.2 การใช้งานพอร์ตเป็นการเอาท์พุต

เมื่อมีการส่งข้อมูลที่มีค่าเป็น 0 ให้กับแต่ละบิตของพอร์ตทุกพอร์ต ข้อมูลนี้จะถูกส่งให้กับฟลิปฟลอปซึ่งจะค้างค่านี้ไว้ และมีผลทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุตนั้นทำงานดังนั้นขาสัญญาณก็จะมีสภาวะลอจิกเป็นลดจิกต่ำด้วย

ส่วนการส่งข้อมูลที่มีค่าเป็น 1 ออกมานั้น ในกรณีที่เป็นการทำงานในแต่ละบิตของพอร์ต 1,2 หรือ 3 จะทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุตนั้นหยุดทำงานมีผลทำให้ขาของสัญญาณเป็นลอจิกสูงด้วยตัวต้านทานที่ Pull-up อยู่ภายในนั้น แต่สำหรับการใช้งานในแต่ละบิตทางทางพอร์ต 0 นั้นจะมีผลแตกต่างออกไป โดยขาสัญญาณจะมีสภาวะอิมพีแดนซ์สูงแทน เนื่องจากไม่มีตัวต้านภายในเชื่อมต่ออยู่นั้นเอง ดังนั้นการใช้งานพอร์ต 0 เป็นการเอาท์พุตข้อมูล จึงจำเป็นต้องใช้ตัวต้านทานภายนอก Pull-up สัญญาณไว้กับลอจิกสูงแทน

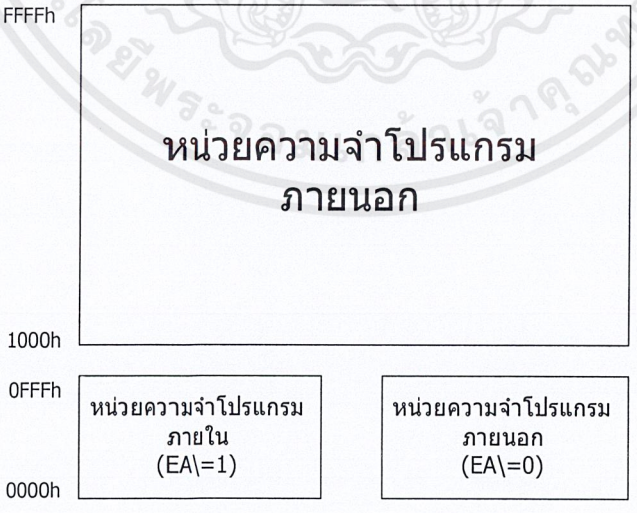
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 หน่วยความจำโปรแกรมของ 8051

หน่วยความจำโปรแกรม(Program Memory)มีไว้เพื่อบรรจุคำสั่งหรือโปรแกรมที่ผู้ใช้พัฒนาขึ้นจัดเก็บไว้ในหน่วยควานจำ โดยอาจจะประกอบอยู่ภายในตัวของไอซี 8051 เอง หรือเป็นไอซีหน่วยความจำ EPROM หรือ ROM แยกออกต่างหากได้ ในกรณีหลังจำเป็นต้องมีการใช้พอร์ตอินพุตเอาท์พุตพอร์ต ทำหน้าที่เป็นบัสแอดเดรสและบัสข้อมูลเพื่อให้สามารถติดต่อกับหน่วยความจำมาตรฐานทั่วไปได้

หน่วยความจำโปรแกรมของ 8051 เป็นบริเวณหน่วยความจำสำหรับเก็บข้อมูลและคำสั่งใช้งานต่างๆ ซึ่งแม้ว่าจะไม่มีการจ่ายกระแสไฟฟ้าให้ระบบข้อมูลเหล่านี้ก็ยังไม่สูญหายโครงสร้างของหน่วยความจำโปรแกรมมีลักษณะเช่นเดียวกับหน่วยความจำที่บรรจุอยู่ในไอซีหน่วยความจำประเภทต่างๆเช่น หน่วยความจำแบบ ROM (Read Only Memory) หรือ EPROM (Erasable Programmable Read Only Memory)

การจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ 8051 จะมีการจัดพื้นที่ดังรูปที่ 3.4 โดยที่ไมโครคอนโทรลเลอร์ 8051 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมสูงสุดได้ไม่เกิน 64 กิโลไบต์และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะตามตำแหน่งของหน่วยความจำนั้น คือ หน่วยความจำโปรแกรมภายใน(Internal Program Memory) ซึ่งเป็นหน่วยความจำ ROM หรือ EPROM ที่อยู่ภายในตัวไอซีไมโครคอนโทรลเลอร์เอง และหน่วยความจำภายนอก (External Program Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ

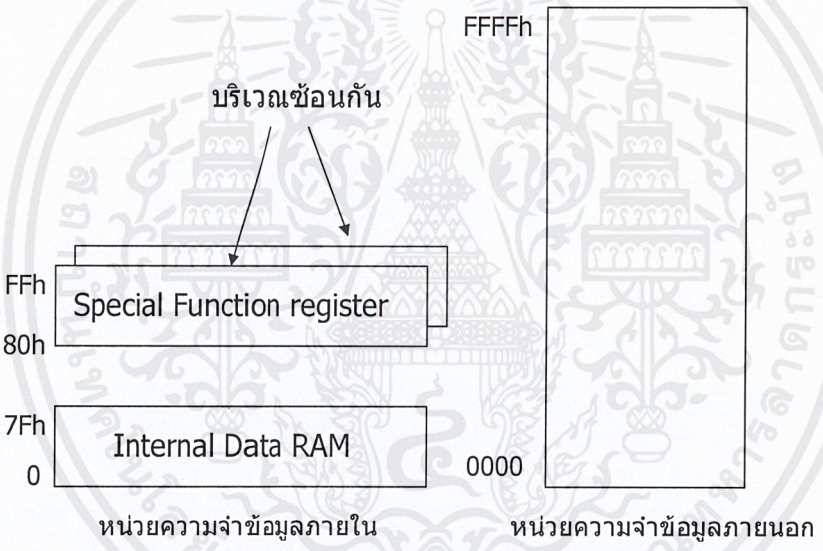


รูปที่3.4 การจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 หน่วยความจำข้อมูลของ 8051

หน่วยความจำข้อมูลมีหน้าที่สำหรับเก็บข้อมูลหรือตัวแปลที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่งคราว โดยที่หน่วยความจำข้อมูลจะมีลักษณะเป็นหน่วยความจำ RAM แบบสแตติก ดังนั้นเมื่อไม่มีการจ่ายไฟให้กับระบบ ก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้ในหน่วยความจำนี้สูญหายไป พื้นที่ของหน่วยความจำข้อมูลของ 8051 สามารถมีได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทออกออกเป็นสองลักษณะตำแหน่งที่ตั้งของหน่วยความจำนั้นดังลักษณะในรูปที่3.5 คือ หน่วยความจำโปรแกรมภายใน (Internal Data Memory)ซึ่งเป็น RAM ที่อยู่ภายในตัวไมโครคอนโทรลเลอร์เอง และหน่วยความจำข้อมูลภายนอก (External Data Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำ RAM มาเพิ่มเติมเข้าไปในวงจร ลักษณะเดียวกับการนำไอซี EPROM มาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง



รูปที่ 3.5 ลักษณะตำแหน่งที่ตั้งของหน่วยความจำข้อมูล

3.5.1 หน่วยความจำข้อมูลภายใน

หน่วยความจำข้อมูลภายใน 8051 มีจำนวนทั้งหมด 256 ไบต์ โดยจำแนกออกได้เป็นสองลักษณะคือ พื้นที่เฉพาะสำหรับตัวประมวลผลกลางใช้งานเท่านั้นซึ่งเรียกว่ารีจิสเตอร์ และพื้นที่ใช้งานทั่วไปสำหรับโปรแกรมใช้งานที่ผู้ใช้สร้างขึ้นมา จากรูปที่ 3.6 แสดงถึงการจัดพื้นที่ของหน่วยความจำข้อมูลภายในของ 8051 ซึ่งแบ่งเป็นสองส่วนดังนี้

3.5.2 หน่วยความจำขนาด 128 ไบต์แรก

บริเวณนี้จะมีตำแหน่งแอดเดรสอยู่ในช่วง 00H-07H ซึ่งแบ่งได้เป็นอีกสามส่วนดังนี้

บริเวณแอดเดรส 00H-1FH จำนวน 32ไบต์จำแนกออกเป็นกลุ่มหรือแบงก์ข้อมูลจำนวน 8 ไบต์รวมทั้งหมดสี่กลุ่ม พื้นที่ข้อมูลในแต่ละกลุ่มจะถูกใช้งานในฐานะของรีจิสเตอร์ทั่วไป เรียกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

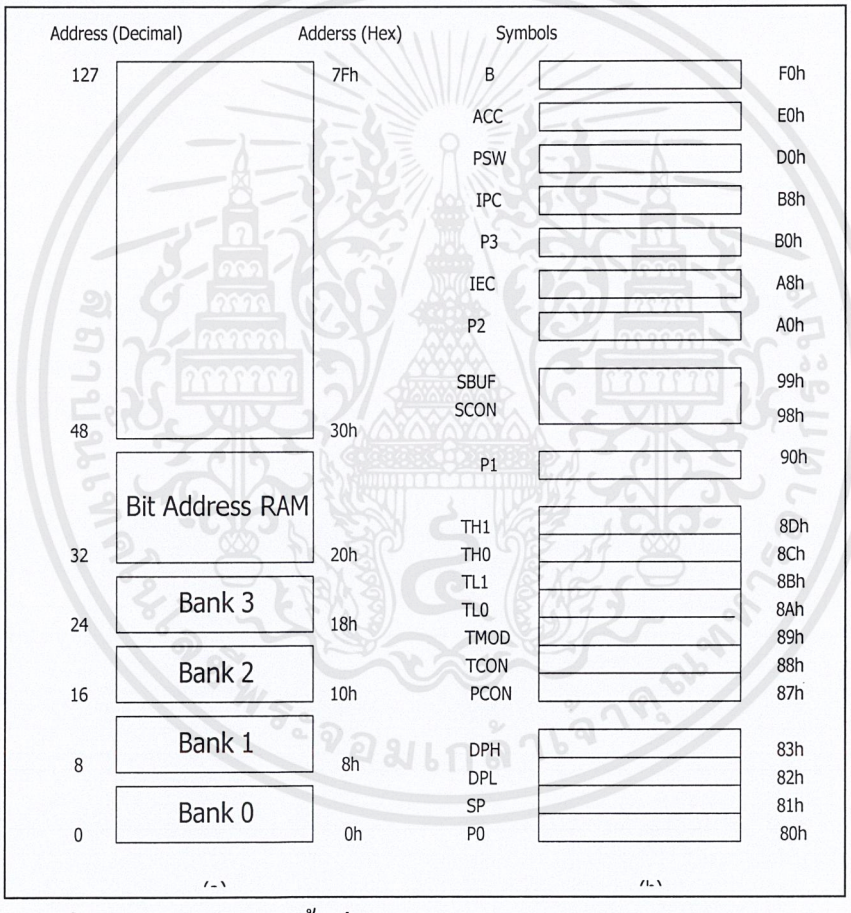
รีจิสเตอร์ R0-R7

บริเวณแอดเดรส 20H-2FH จำนวน16 ไบต์ จะเป็นพื้นที่ส่วนสำหรับผู้ใช้ที่สามารถอ้างถึงข้อมูลได้ทั้งแบบไบต์และแบบบิต

บริเวณแอดเดรส 30H-7FH เป็นบริเวณที่สาสารถนำไปใช้งานได้อย่างอิสระ โดยสามารถอ้างอิงได้เฉพาะในลักษณะของไบต์ข้อมูลตามปกติเท่านั้น

3.5.3 หน่วยความจำขนาด 128 ไบต์ถัดไป

พื้นที่ตั้งแต่แอดเดรส 80H-FFH เป็นหน่วยความจำที่นำมาใช้งานเป็นรีจิสเตอร์หน้าที่พิเศษ แต่ก็ยังมีบริเวณของหน่วยความจำที่อยู่บริเวณนี้ที่ผู้ใช้สามารถเก็บข้อมูลได้ แต่การเรียกใช้งานจะต้องมีการเข้าถึงข้อมูลแบบโดยอ้อมเท่านั้น



รูปที่ 3.6 แสดงถึงการจัดพื้นที่ของหน่วยความจำข้อมูลภายในของ 8051

3.6 รีจิสเตอร์หน้าที่พิเศษ

เป็นรีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของอุปกรณ์หรือพอร์ตของ 8051 ทั้งหมด โดยมีตำแหน่งอยู่ในบริเวณแอดเดรส 80H-FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุถึงชื่อรีจิสเตอร์หรือตำแหน่งแอดเดรสที่เป็นของรีจิสเตอร์นั้นก็ได้

3.6.1 แอกคิวมูเลเตอร์ (Accumulator)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นรีจิสเตอร์ขนาด 8 บิตทำหน้าที่เก็บข้อมูลที่จะส่งให้หน่วยทำงานในซีพียูและเก็บผลลัพท์ที่ได้จากการทำงานนั้น การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ A

3.6.2 รีจิสเตอร์ B

เป็นรีจิสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณหารตัวเลข ในกรณีที่ไม่ใช้การคำนวนทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจิสเตอร์ทั่วไปได้

3.6.3 โปรแกรมเคาน์เตอร์ (Program Counter)

เป็นรีจิสเตอร์ที่ใช้สำหรับการชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม ซึ่งจะต้องไปทำงานในลำดับถัดไป การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ PC

3.6.4 สแต็กพอยน์เตอร์(Stack Pointer)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งของตัวชี้หรือพอยน์เตอร์ของบริเวณสแต็กสำหรับเก็บข้อมูลแอกคิวมูเลเตอร์ รีจิสเตอร์ต่างๆ รวมทั้งข้อมูลจากโปรแกรม ค่าเริ่มต้นของสแต็กจะอยู่ที่ตำแหน่ง 07H การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ SP

3.6.5 ตัวชี้ข้อมูลหรือดาต้าพอยน์เตอร์(Data Pointer)

เป็นรีจิสเตอร์ขนาด 16 บิตซึ่งเรียกว่ารีจิสเตอร์DPTR และสามารถใช้งานแยกออกเป็นรีจิสเตอร์ขนาด 8 บิตสองตัวคือ รีจิสเตอร์ DPH และ DPL เพื่อเก็บค่าแอดเดรสของหน่วยความจำที่จะต้องใช้งานภายในโปรแกรมหรืออาจเป็นแอดเดรสของอุปกรณ์ภายนอก

3.6.6 โปรแกรมสเตตัสเวิร์ด(PSW)

รีจิสเตอร์นี้ทำหน้าที่บอกถึงแฟล็กสภาวะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงก์(Bank) ของรีจิสเตอร์ที่ใช้งานด้วย

3.6.7 รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ต(Port Register)

รีจิสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ตอินพุตเอาท์พุตโดยตรงซึ่งจะเป็นรีจิสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะการอินพุตหรือการเอาท์พุตข้อมูลได้

3.6.8 รีจิสเตอร์ SBUF

เป็นบัฟเฟอร์ขนาด 8 บิตสำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและการส่งข้อมูล

3.6.9 รีจิสเตอร์ PCON

เป็นรีจิสเตอร์ควบคุมการทำงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโปรเซสเซอร์ การกำหนดอัตราทวีคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรมและแฟล็กสภาวะการทำงานทั่วไป

3.6.10 รีจิสเตอร์ IP,IE,TMOD,SCON

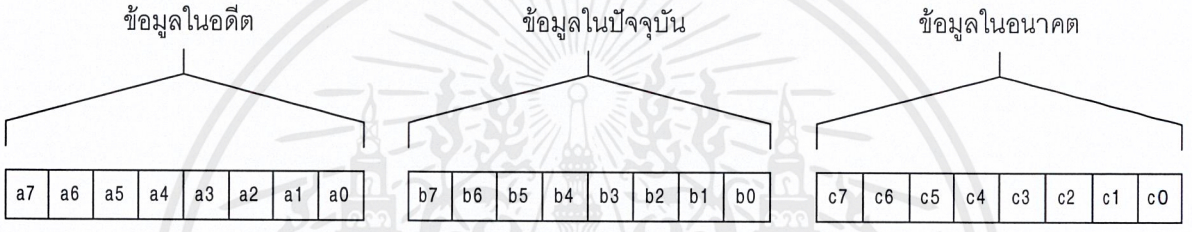
เป็นกลุ่มรีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของการอินเตอร์รัปต์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การออกแบบอัลกอลิทึม

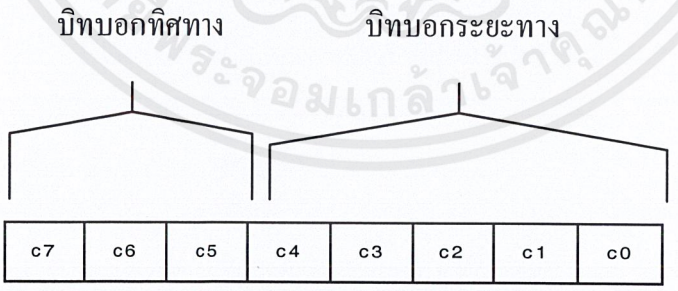
4.1 การออกแบบอินพุทและเอาท์พุทของระบบแยกแยะความแตกต่าง

การออกแบบอินพุทให้กับระบบแยกแยะความแตกต่าง จะออกแบบโดยใช้ข้อมูลจำนวน 3 ชุด ข้อมูลชุดที่ 1 คือเส้นทางในอดีต(t-1) ข้อมูลชุดที่ 2 คือเส้นทางปัจจุบัน (t) และข้อมูลชุดที่ 3 คือเส้นทางในอนาคต(t+1) แสดงในรูปที่ 4.1 โดยเราจะนำข้อมูลทั้ง 3 ชุดมาสร้าง Lookup-table ให้กับระบบแยกแยะความแตกต่าง



รูปที่ 4.1 กลุ่มข้อมูลอินพุทของระบบแยกแยะความแตกต่าง

โดยที่ข้อมูลแต่ละชุดจะมีขนาด 8 บิท ซึ่งข้อมูล 3 บิทแรกของแต่ละชุดจะเป็นทิศทางการเคลื่อนที่ของหุ่นยนต์ และข้อมูล 5 บิทหลังจะเป็นระยะทางที่หุ่นยนต์เคลื่อนที่ไปแสดงในรูปที่ 4.2

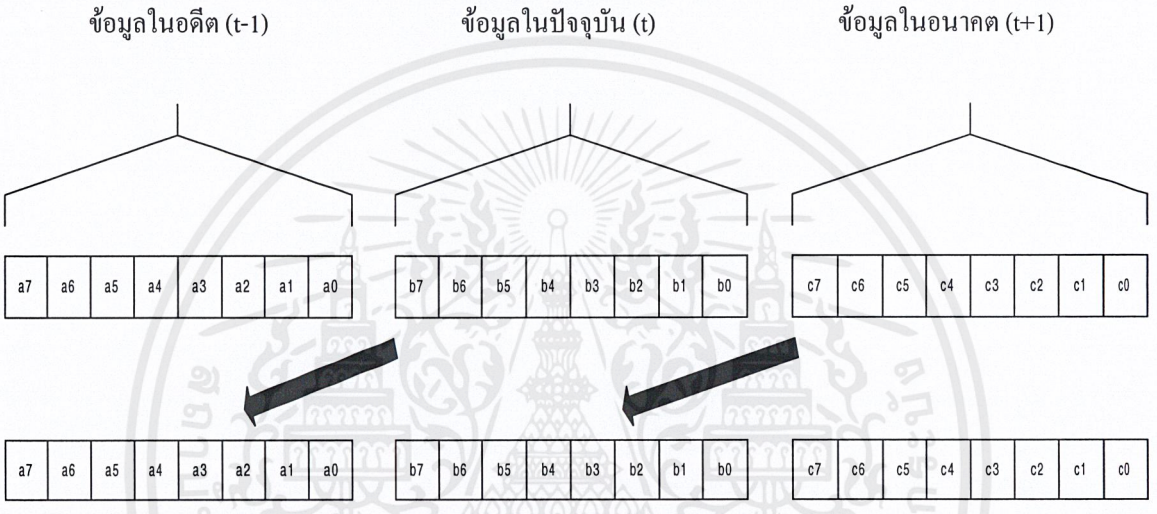


รูปที่ 4.2 ข้อมูลในแต่ละเหตุการณ์

เมื่อหุ่นยนต์เคลื่อนที่และเปลี่ยนเส้นทาง 1 ครั้งจะนับเป็น 1 เส้นทาง เมื่อหุ่นยนต์มีการเคลื่อนที่เปลี่ยนทิศทางทั้งหมด 3 ครั้งแล้ว ก็จะมีการกำหนดอินพุทและเอาท์พุทให้ระบบแยกแยะความแตกต่างทั้ง 8 ตัว 1ครั้ง หลังจากที่กำหนดอินพุทและเอาท์พุทให้ระบบแยกแยะความแตกต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้ว หุ่นยนต์ก็จะวิ่งไปเก็บข้อมูลเส้นทางใหม่ เมื่อเก็บข้อมูลได้แล้วก็จะเลื่อนข้อมูลไปทางซ้าย 1 ครั้งโดยจะเลื่อนข้อมูลของเส้นทางปัจจุบัน(t) ไปแทนที่ข้อมูลของเส้นทางในอดีต(t-1) และข้อมูลของเส้นทางในอนาคต(t+1) ก็จะไปแทนที่ข้อมูลของเส้นทางปัจจุบัน(t) และนำข้อมูลของเส้นทางที่เก็บได้จากการเดินทางครั้งใหม่สุดมาเป็นข้อมูลของเส้นทางในอนาคต(t+1) จากนั้นก็จะนำข้อมูลทั้ง 3 ชุดที่เก็บมาใหม่มากำหนดเป็นอินพุทและเอาท์พุทให้ระบบแยกแยะความแตกต่างอีกครั้ง และจะทำการเก็บข้อมูลในลักษณะนี้ไปจนกว่าหุ่นยนต์จะเคลื่อนที่กลับมาที่จุดเริ่มต้น ก็จะเป็นการเรียนรู้เส้นทางครบ 1 รอบ การเลื่อนของข้อมูลจะแสดงดังรูปที่ 4.3



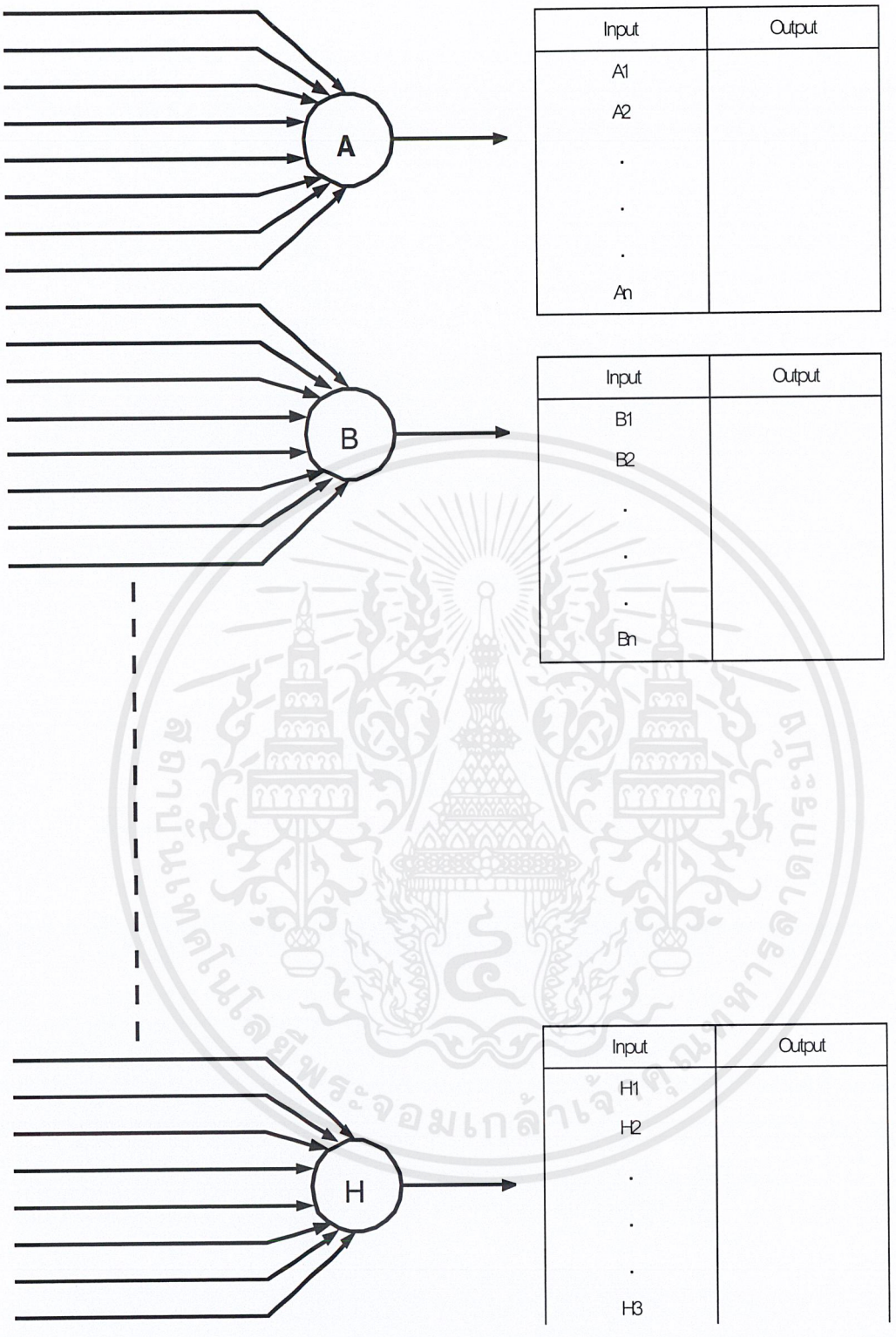
รูปที่ 4.3 แสดงการเลื่อนของข้อมูลเมื่อช่วงเวลาเปลี่ยนไป

4.2 การสร้าง Lookup-table ของระบบแยกแยะความแตกต่างในสถาวะรีคอกไนซ์(Recognize)

ระบบแยกแยะความแตกต่างมีทั้งหมด 8 ตัวทุกตัวจะมีLookup-table เป็นของตัวเอง ขนาดของLookup-table จะมีขนาดเล็กหรือใหญ่ขึ้นอยู่กับข้อมูลที่เรานำมาสอนให้กับระบบแยกแยะความแตกต่าง(ในที่นี้คือเส้นทาง)และLookup-tableจะมีขนาดใหญ่ขึ้นกว่าเดิมเมื่อเราได้สอนให้ให้ระบบแยกแยะความแตกต่างเพิ่มขึ้น ระบบแยกแยะความแตกต่างแต่ละตัวจะมีอินพุท 8 บิทและเอาท์พุท 1 บิท จากหัวข้อที่แล้วเราได้ข้อมูลของเส้นทางมา 3 ชุดแรก เราจะนำมากำหนดเป็นอินพุทและเอาท์พุทให้ระบบแยกแยะความแตกต่างแต่ละตัว โดยข้อมูลของเส้นทางในอดีต(t-1)และข้อมูลของเส้นทางปัจจุบัน(t)จะถูกกำหนดให้เป็นเป็นอินพุทให้ระบบแยกแยะความแตกต่างแต่ละตัว บิทข้อมูลของเส้นทางในอนาคต(t+1)จะถูกกำหนดให้เป็นเป็นเอาท์พุทให้ระบบแยกแยะความแตกต่างแต่ละตัว แสดงดังรูปที่ 4.5 และรูปที่4.6 แสดงตารางการกำหนดอินพุทและเอาท์พุทให้ระบบแยกแยะความแตกต่างแต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

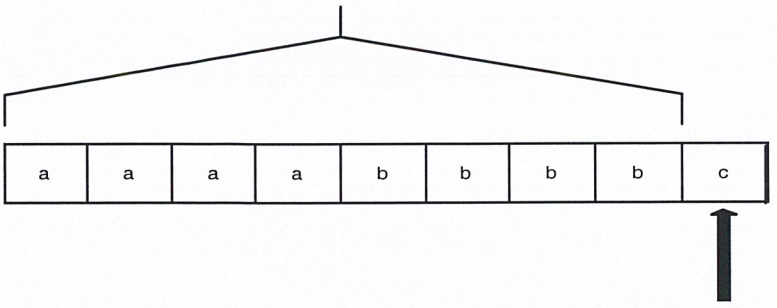
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 4.4 แสดงแบบจำลองระบบแยกแยะความแตกต่างของการเก็บข้อมูลไว้ในLookup-table ของระบบแยกแยะความแตกต่างแต่ละตัว

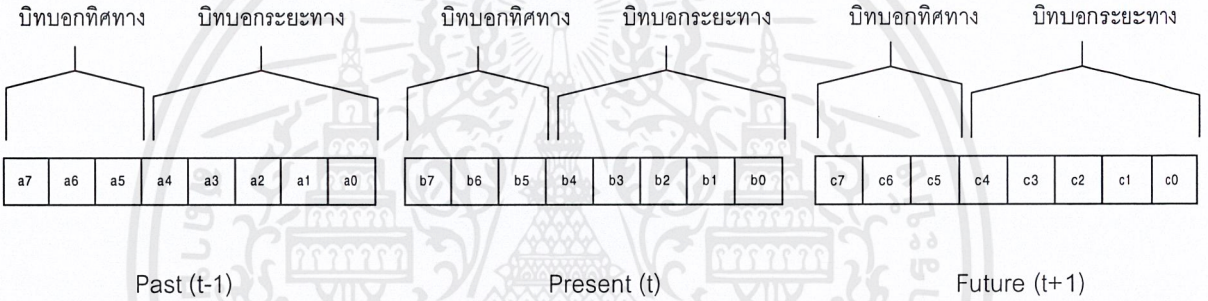
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
46271

ข้อมูลที่มาจากสถานะอดีตและสถานะปัจจุบัน



ข้อมูลที่ได้มาจากอนาคต

รูปที่ 4.5 แสดงข้อมูลที่ถูกกำหนดให้เป็นอินพุทและเอาท์พุทให้ระบบแยกแยะความแตกต่างแต่ละตัว



Neural	Input Neural	Output Neural
A	a7 a5 a2 b7 b6 b5 b2 b0	c7
B	a6 a5 a2 a0 b7 b6 b5 b3	c6
C	a7 a6 a5 a1 b6 b5 b3 b2	c5
D	a7 a6 a4 a0 b6 b5 b2 b1	c4
E	a6 a5 a3 a1 b7 b6 b4 b2	c3
F	a7 a6 a3 a0 b6 b5 b4 b1	c2
G	a6 a5 a4 a2 b7 b6 b3 b0	c1
H	a7 a5 a4 a3 b6 b5 b2 b0	c0

รูปที่4.6 แสดงตารางการกำหนดอินพุทและเอาท์พุทให้ระบบแยกแยะความแตกต่างแต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

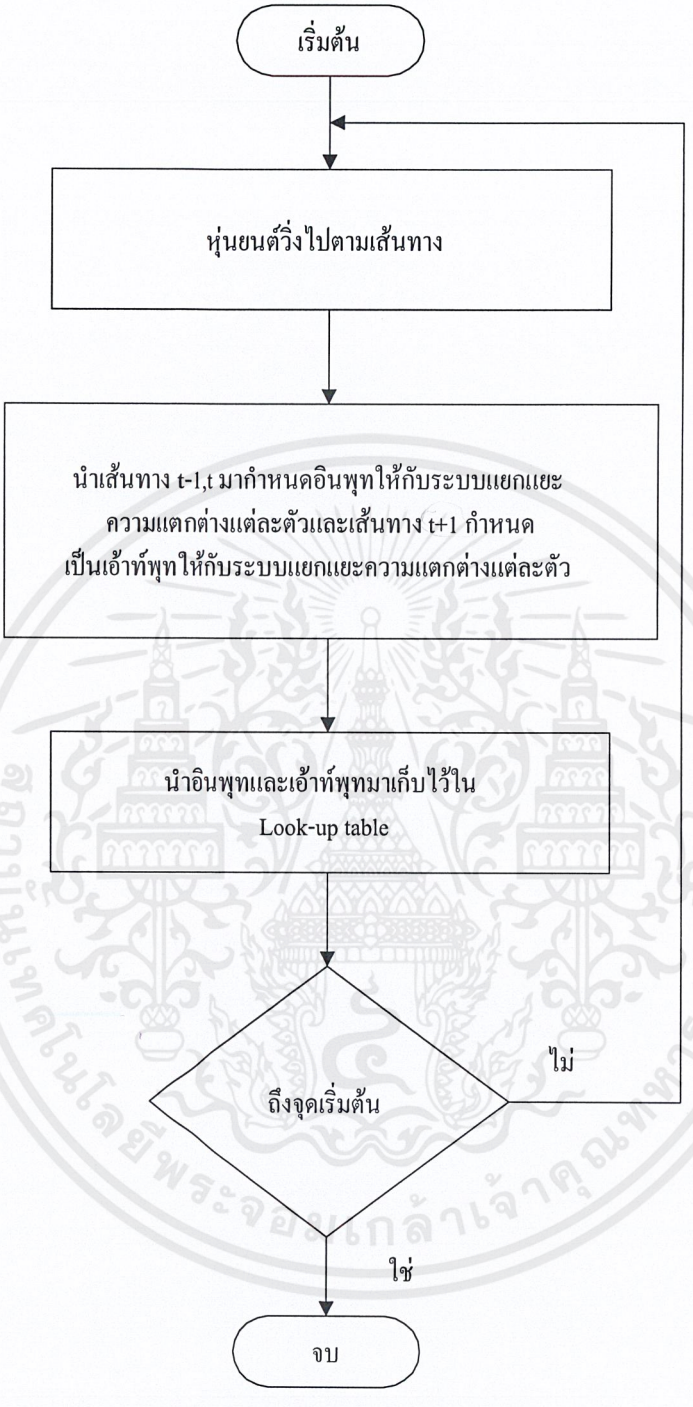
4.3 การค้นหาเส้นทาง

การค้นหาเส้นทางก็คือสภาวะการรีคอล(Recall) ของระบบแยกแยะความแตกต่าง เราสามารถทำได้โดยการป้อนข้อมูลเริ่มต้นให้กับระบบแยกแยะความแตกต่าง 2 ชุด คือ ข้อมูลเส้นทางในอดีต(t-1) และข้อมูลเส้นทางปัจจุบัน(t) ข้อมูลทั้ง 2 ชุดนี้จะถูกนำไปกำหนดเป็นอินพุทให้กับระบบแยกแยะความแตกต่างแต่ละตัว หลังจากนั้นระบบแยกแยะความแตกต่างแต่ละตัวก็จะสร้างเอาท์พุทออกมาแล้วนำมารวมกันเป็นข้อมูลของเส้นทางในอนาคต(t+1) โดยเอาค่ามาจาก Lookup-table ที่ได้เรียนรู้ไปแล้วตั้งแต่ตอนแรก การสร้างข้อมูลเริ่มต้นให้กับระบบแยกแยะความแตกต่างสามารถทำได้โดยการกำหนดเส้นทางเริ่มต้นเป็นระยะทางสั้นๆให้หุ่นยนต์ (ต้องเป็นเส้นทางที่หุ่นยนต์เคยเรียนรู้มาก่อน) หลังจากนั้นหุ่นยนต์ก็จะเคลื่อนที่ต่อจากเส้นทางที่เริ่มได้เองโดยไม่ต้องมีเส้นทางอีกต่อไป

4.3.1 การตัดสินใจเลือกเส้นทาง

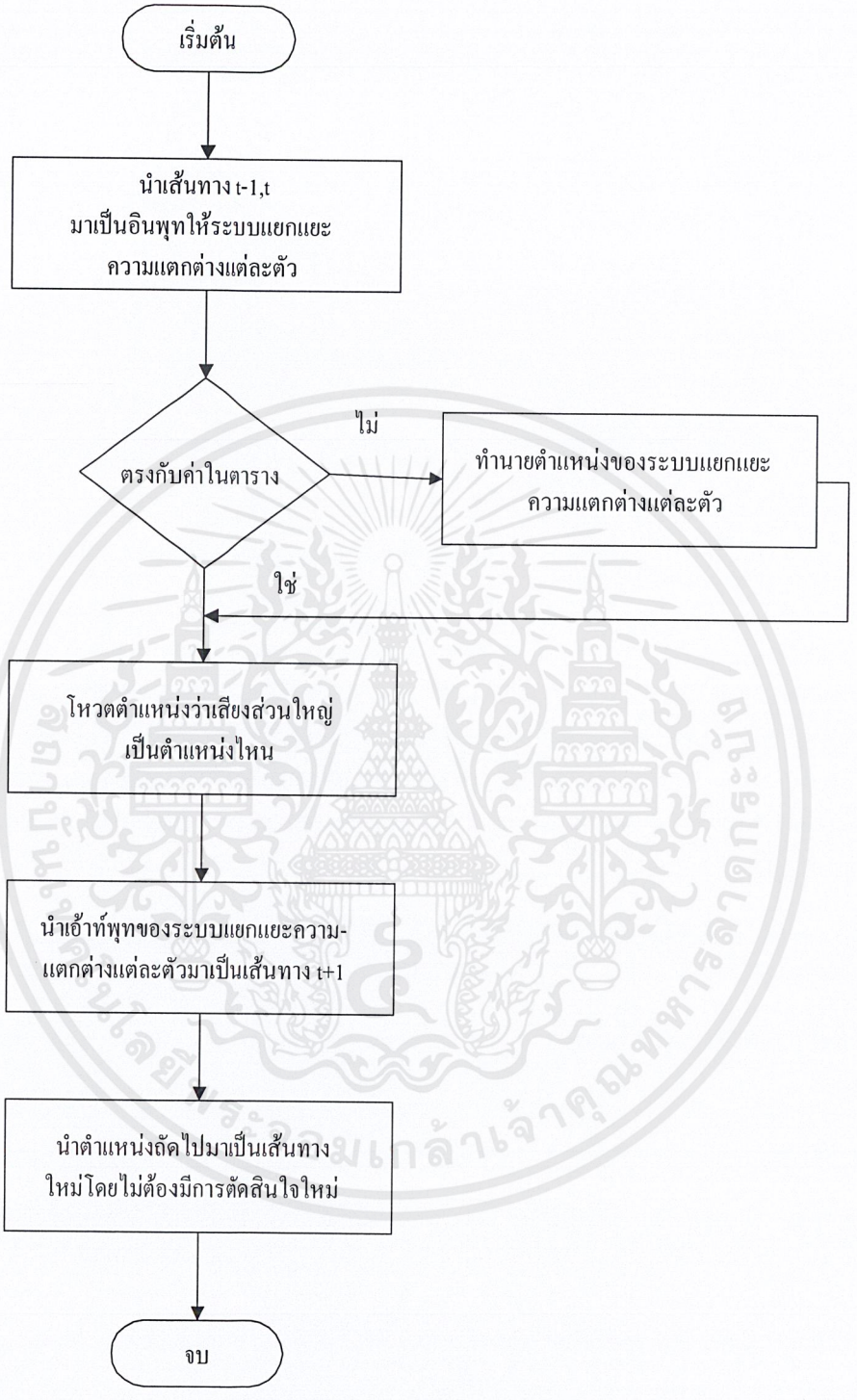
ก่อนที่หุ่นยนต์จะสามารถเคลื่อนที่ต่อจากเส้นทางที่เราเริ่มต้นให้ได้นั้น จะต้องมีกระบวนการในการตัดสินใจว่าเส้นทางที่จะเคลื่อนที่ต่อไปนั้นควรจะเป็นเส้นทางไหน ซึ่งเป็นการนำเอาข้อมูลของเส้นทางในอดีต(t-1) และข้อมูลของเส้นทางในปัจจุบันมาพิจารณาซึ่งมีอยู่ 4 วิธี แสดงในรูปที่ 4.7,4.8,4.9,4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



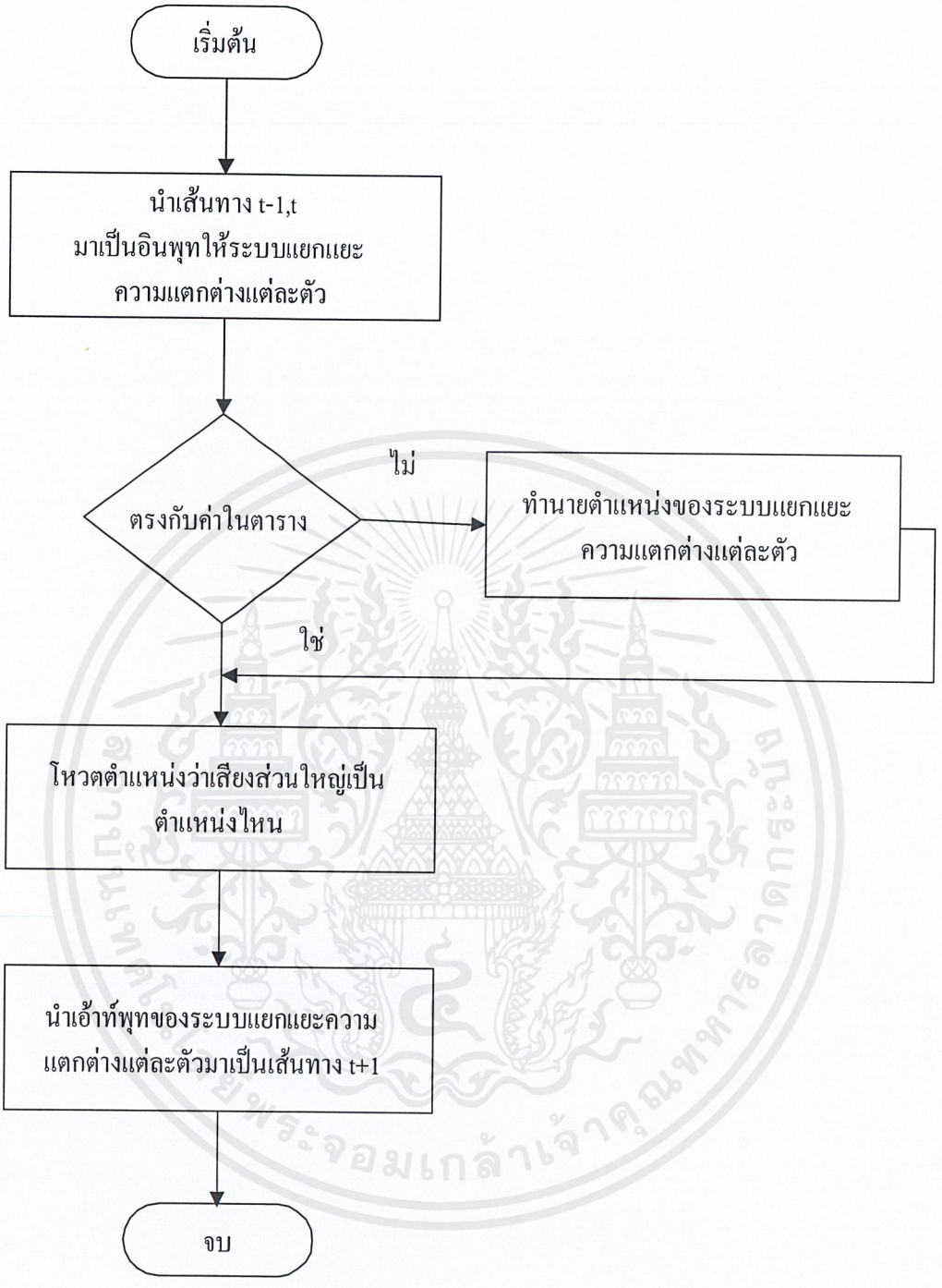
รูปที่ 4.7 แผนผังของช่วงการจำเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



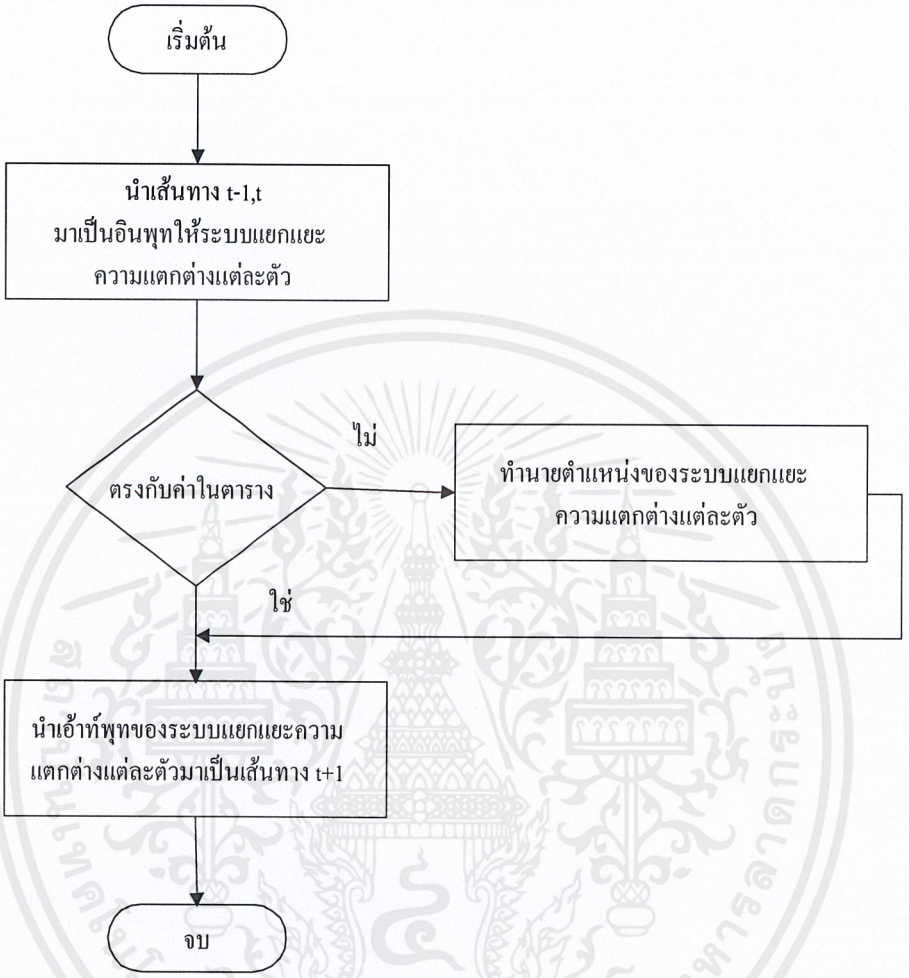
รูปที่ 4.8 แผนผังของการรีคอลวิธีที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



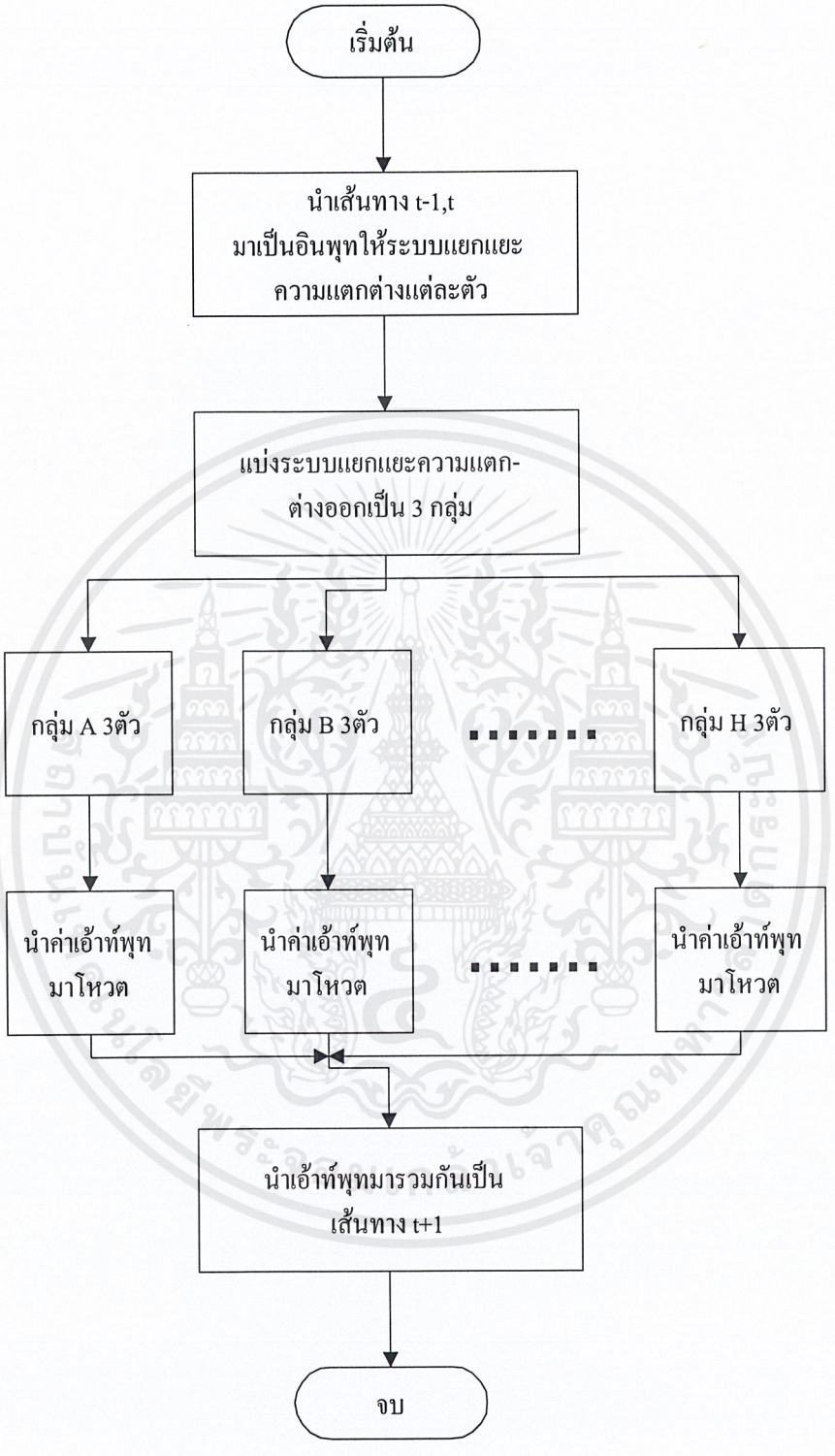
รูปที่ 4.9 แผนผังของการรีคอลวิธีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แผนผังของการรีคอลวิธีที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แผนผังของการรีคอลวิธีที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
หลักการออกแบบวงจร

5.1 โครงสร้างฮาร์ดแวร์(Hardware)

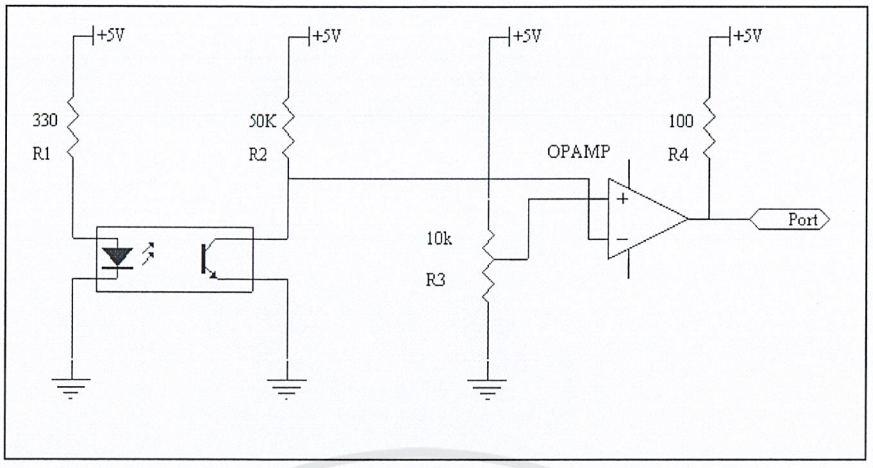
การทำงานของโรบอทจะถูกควบคุมด้วยไมโครคอนโทรลเลอร์ MCS-51 ซึ่งเป็นตัวควบคุมการทำงานทั้งหมด โดยที่วงจรทั้งหมดของระบบจะแสดงไว้ดังรูปที่ 5.3 ส่วนประกอบวงจรการทำงานของโรบอทจะประกอบด้วยวงจรส่วนต่างๆดังนี้

1. ส่วนของวงจรเซ็นเซอร์
2. ส่วนของวงจรขับมอเตอร์
3. ส่วนของการควบคุมการทำงานโดยใช้ไมโครคอนโทรลเลอร์ MCS-51

5.1.1 วงจรเซ็นเซอร์

ในวงจรภาคเซ็นเซอร์เราจะใช้โฟโต้ไดโอดเป็นตัวส่งแสงอินฟาเรดและโฟโต้ – ทรานซิสเตอร์เป็นตัวรับ โดยที่เอาท์พุทขาคอลเลคเตอร์ของโฟโต้ทรานซิสเตอร์จะถูกต่อเข้ากับขาอินเวิสก์ของออปแอมป์ และขานอนอินเวิสก์ของออปแอมป์จะต่อกับความต้านทานปรับค่าได้ การจัดวงจรของออปแอมป์จะจัดวงจรแบบเปรียบเทียบแรงดัน ถ้าตัวเซ็นเซอร์อินฟาเรดอยู่ตำแหน่งของเส้นทาง(ตำแหน่งของเส้นดำ)ก็จะทำให้เอาท์พุทของออปแอมป์เป็นลอจิก “ 0 “ และถ้าตัวเซ็นเซอร์อินฟาเรดอยู่ตำแหน่งที่ออกนอกเส้นทางก็จะทำให้เอาท์พุทของออปแอมป์เป็นลอจิก “ 1 “โดยที่ตัวเซ็นเซอร์3ตัวจะอยู่ที่ด้านหน้าของรถและบริเวณด้านหน้าของล้อทั้งสองข้าง เอาท์พุทของวงจรเซ็นเซอร์ที่เป็นตัวแสดงทิศทางการเคลื่อนที่ของโรบอท จะถูกนำไปเป็นอินพุทของพอร์ท P1.0 P1.1 และ P1.2

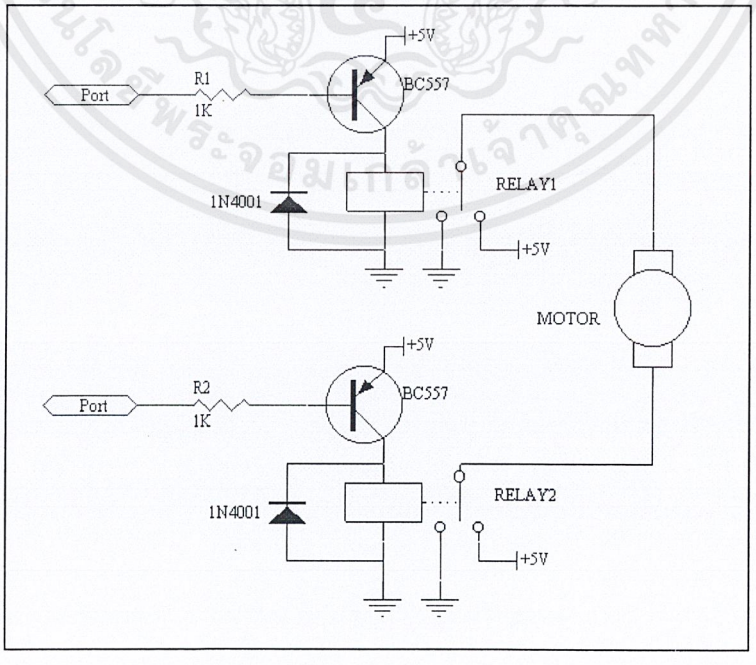
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 วงจรเซ็นเซอร์

5.1.2 วงจรขับมอเตอร์

วงจรขับมอเตอร์ของโรบอทจะใช้ทรานซิสเตอร์ชนิดพีเอ็นพี (PNP)เป็นตัวควบคุมการทำงานของรีเลย์ โดยที่ขาเบส(Base)ของทรานซิสเตอร์จะถูกต่อกับพอร์ทของไมโครคอลโทรล - เลอร์ และขาคอลเลคเตอร์(Collector)ของทรานซิสเตอร์จะต่ออยู่กับรีเลย์ มอเตอร์ 1 ตัวจะถูกควบคุมโดยรีเลย์ 1 คอนแทค 2 ตัว โดยการหมุนของมอเตอร์จะเป็นได้ 3 ลักษณะคือ หมุนซ้าย หมุนขวา และหยุดหมุน



รูปที่5.2 วงจรขับมอเตอร์

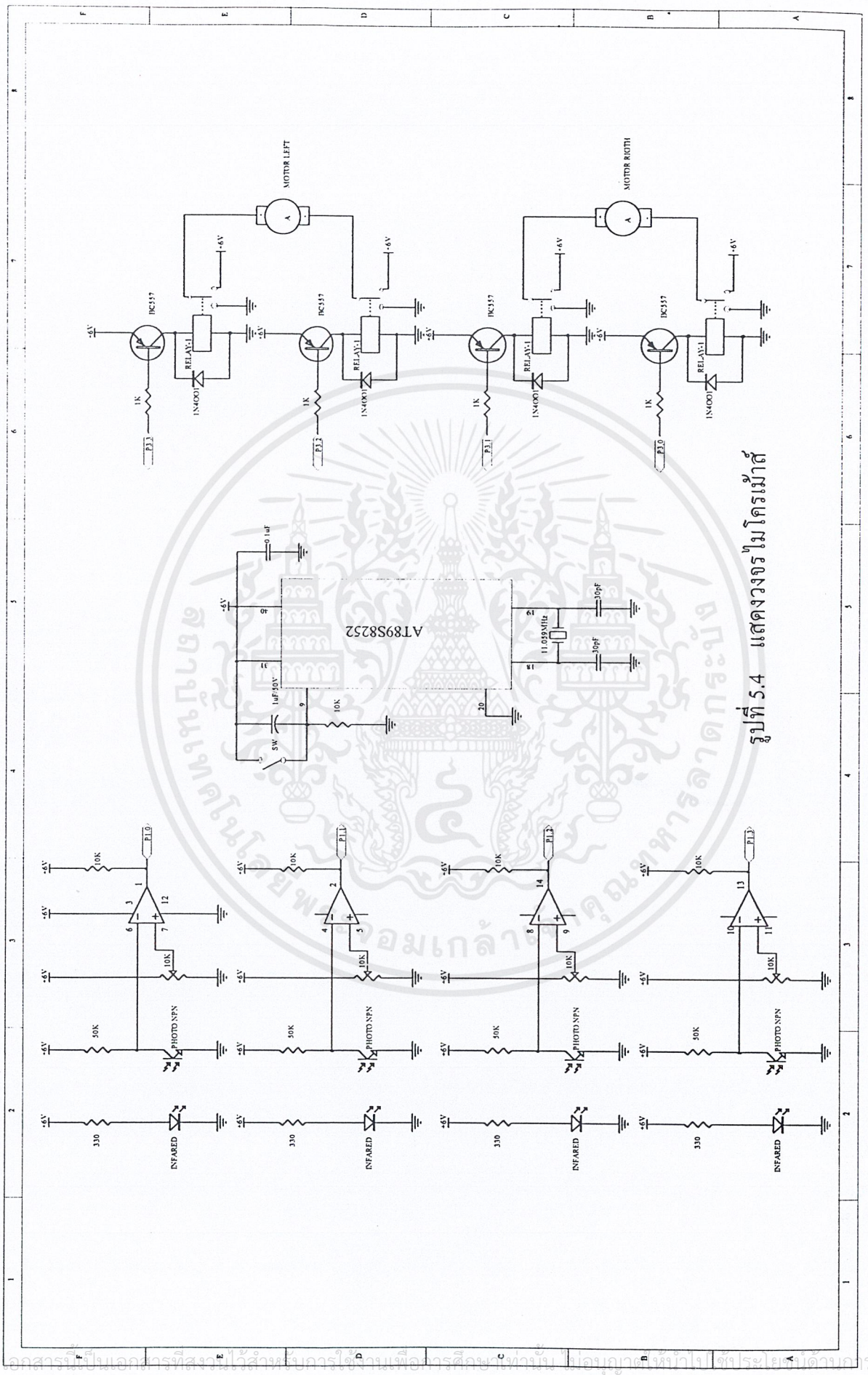
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.3 ส่วนของการควบคุมการทำงานโดยใช้ไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์จะมีหน้าที่ควบคุมการทำงานของโรบอททั้งหมด ไมโครคอนโทรลเลอร์จะมีหน้าที่การทำงาน 4 หน้าที่คือ

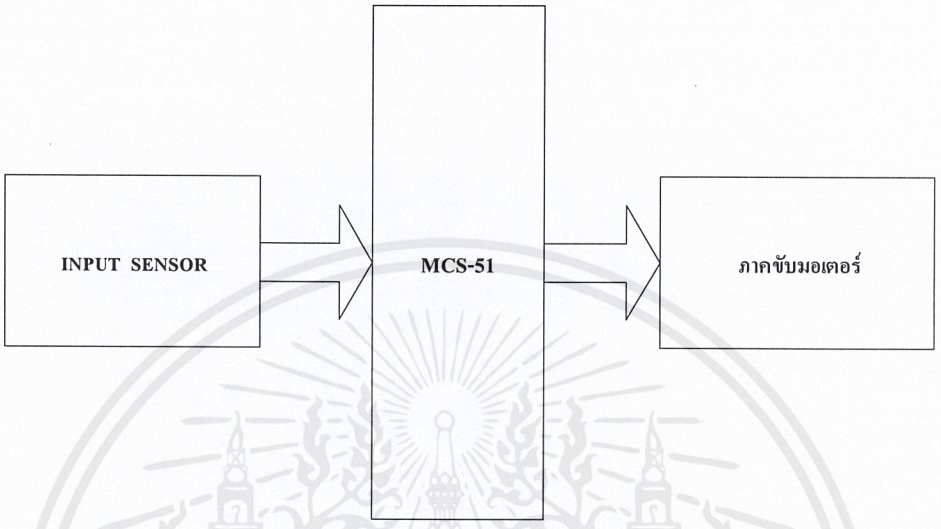
1. ตรวจสอบสถานะของตัวเซ็นเซอร์ทิศทาง
2. นับเวลาการเคลื่อนที่ของโรบอทเพื่อนำไปคำนวนเป็นระยะทาง
3. ควบคุมการหมุนของมอเตอร์
4. ทำการประมวลผลการทำงานของระบบโครงข่ายแบบฉลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 แสดงวงจรไมโครเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 แสดงบล็อกไดอะแกรมของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การทดลองและบทสรุป

6.1 การทดลอง

การทดลองหุ่นยนต์จดจำเส้นทางจะเริ่มต้นโดยการนำปากกาสีดำมาลากเป็นเส้นทางบนกระดานสีขาวเพื่อเป็นเส้นทางให้หุ่นยนต์เดินทาง

การทดลองที่ 1

โปรแกรมที่ใช้ในการทดสอบจะมีการทำงานเหมือนกับแผนผังการทำงานในรูปที่ 4.9 โดยการเริ่มต้นเราจะให้หุ่นยนต์วิ่งไประยะทางหนึ่งก็จะนำเส้นทาง t-1 และ t มากำหนดเป็นอินพุทให้กับนิวรอลแต่ละตัว จากนั้นนิวรอลก็จะนำอินพุทของตัวเองไปเก็บในตารางLook-up table ที่ได้จากการเรียนรู้ว่าตำแหน่งอินพุทที่เข้ามาของตัวเองอยู่ตรงกับตำแหน่งที่เท่าไร จากนั้นก็จะทำการโหวตว่าเสียงส่วนใหญ่เป็นตำแหน่งที่เท่าไร หลังจากนั้นก็จะไปเอาคำตอบที่ตำแหน่วที่โหวตได้มาเป็นคำตอบ ซึ่งคำตอบที่ได้ก็คือเส้นทางที่จะวิ่งต่อไป(t+1) หุ่นยนต์ก็จะวิ่งไปตามเส้นทางที่ได้จากการโหวต หลังจากนั้นก็จะทำการโหวตเส้นทางใหม่อีกครั้งและทำการโหวตไปเรื่อยๆ เพื่อการวิ่งให้ได้เส้นทางเดิมที่เคยจดจำมา

ผลการทดลอง

1. ถ้าเส้นทางที่หุ่นยนต์วิ่งมีความแตกต่างของระยะทางและทิศทางมาก หุ่นยนต์จะวิ่งได้ค่อนข้างใกล้เคียงกับระยะทางเดิมที่จำ ถึงแม้จะมีระยะทางไกล
2. ถ้าเส้นทางที่หุ่นยนต์วิ่งมีความแตกต่างของระยะทางและทิศทางน้อย หุ่นยนต์จะวิ่งไม่ใกล้เคียงกับระยะทางเดิมที่จำ และมีระยะทางเล็กลงด้วย เช่นถ้าเส้นทางเดิมเป็นวงใหญ่ หุ่นยนต์ก็จะวิ่งเป็นวงเล็กลงและเลื่อนตำแหน่งออกไป

การทดลองที่ 2

โปรแกรมที่ใช้ในการทดลองจะมีการทำงานเหมือนกับแผนผังการทำงานในรูป4.8 มีลักษณะการทำงานเหมือนการทดลองที่ 1 แต่จะต่างกันที่ไม่มีการโหวตตำแหน่งของเส้นใหม่ทุกๆครั้ง แต่จะโหวตครั้งเดียวแล้วนำเอาตำแหน่งถัดไปของเส้นทางที่โหวตได้มาเป็นเส้นทางใหม่เลย

ผลการทดลอง

ถ้าการโหวตตำแหน่งเริ่มต้นของเส้นทางใหม่ถูกต้อง หุ่นยนต์ก็จะสามารถวิ่งต่อไปจนครบเส้นทางเดิมได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3

โปรแกรมที่ใช้ในการทดลองจะมีการทำงานเหมือนกับแผนผังการทำงานในรูป4.10 มีลักษณะการทำงานเริ่มต้นเหมือนการทดลองที่ 1 แต่จะต่างกันที่ไม่มีการโหวตตำแหน่งของเส้นทาง กล่าวคือเมื่อนิวรอลแต่ละตัวได้อินพุทมาเป็นตำแหน่งไหน ก็จะนำค่าของเอ้าท์พุทของตำแหน่งนั้นมาเป็นคำตอบเลยโดยไม่สนใจว่านิวรอลตัวอื่นๆจะให้ค่าเป็นตำแหน่งใด

ผลการทดลอง

เราพบว่าหุ่นยนต์วิ่งไปเป็นเส้นทางใหม่ๆเสมอ ซึ่งไม่ใช่เส้นทางที่เคยเรียนรู้มาก่อน

6.2 สาเหตุของปัญหา

1. เนื่องจากจำนวนบิตข้อมูลที่นำมาเป็นอินพุทของนิวรอลมีจำนวนน้อยเกินไป
2. เนื่องจากจำนวนนิวรอลที่ใช้มีจำนวนน้อยเกินไป โดยการทดลองนี้ใช้นิวรอล 8 ตัว
3. จากสาเหตุที่ 1 และ 2 จึงไม่สามารถแยกแยะความแตกต่างของเส้นทางที่มีลักษณะคล้ายกันหรือใกล้เคียงกันได้

6.3 แนวทางการแก้ไขปัญหาและพัฒนา

1. เพิ่มจำนวนบิตข้อมูลที่นำมาเป็นอินพุทของนิวรอล เช่นเพิ่มจำนวนเซ็นเซอร์
2. เพิ่มจำนวนนิวรอล
3. เพิ่มจำนวนอินพุทให้กับนิวรอล

6.4 สรุปผลการทดลอง

จากการทดลองหลังจากที่หุ่นยนต์ได้เรียนรู้จดจำเส้นทางแล้ว เมื่อเราให้หุ่นยนต์ทำงานในสภาวะรีคอลเราจะพบว่าหุ่นยนต์จะสามารถเดินทางตามเส้นทางที่ได้เรียนรู้ไปแล้วได้ แต่ถ้าเส้นทางที่นำมาให้หุ่นยนต์จดจำมีลักษณะใกล้เคียงกันมาก จะเกิดความผิดพลาดคือบางครั้งหุ่นยนต์จะเดินทางเป็นเส้นทางที่สั้นกว่าเดิมหรือเดินทางออกนอกเส้นทางไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RY_1                BIT     P2.0
RY_2                BIT     P2.1
RY_3                BIT     P2.2
RY_4                BIT     P2.3
LEFT_LED            BIT     P2.4
RIGHT_LED           BIT     P2.5
START_BUTTOM        BIT     P2.6
RECALL_MODE         BIT     P2.7
SENSORS             EQU     P1
NEURAL_A            EQU     20H
NEURAL_B            EQU     21H
NEURAL_C            EQU     22H
NEURAL_D            EQU     23H
NEURAL_E            EQU     24H
NEURAL_F            EQU     25H
NEURAL_G            EQU     26H
NEURAL_H            EQU     27H
TEMP_A              EQU     28H
TEMP_B              EQU     29H
TEMP_C              EQU     2AH
TEMP_D              EQU     2BH
TEMP_E              EQU     2CH
TEMP_F              EQU     2DH
TEMP_G              EQU     2EH
TEMP_H              EQU     2FH
EVEN_POINTER_1      EQU     30H
EVEN_POINTER_2      EQU     31H
ODD_POINTER_1       EQU     32H
ODD_POINTER_2       EQU     33H
PAST                EQU     34H
PRESENT             EQU     35H
FUTURE              EQU     36H
TIME_BUF1           EQU     37H
TIME_BUF2           EQU     38H
TIME_BUF3           EQU     39H
NEURAL_BUFF         EQU     3AH
MIN                 EQU     3BH
MAX                 EQU     3CH
COUNT_1             EQU     3DH
TEMP_0              EQU     3EH;DPL
TEMP_1              EQU     3FH;DPH
AN_2                EQU     40H
AN_1                EQU     41H
BN_2                EQU     42H
BN_1                EQU     43H
CN_2                EQU     44H
CN_1                EQU     45H
DN_2                EQU     46H
DN_1                EQU     47H
EN_2                EQU     48H
EN_1                EQU     49H
FN_2                EQU     4AH
FN_1                EQU     4BH
GN_2                EQU     4CH
GN_1                EQU     4DH
HN_2                EQU     4EH
HN_1                EQU     4FH
PAST_BUFF           EQU     50H
PRESENT_BUFF        EQU     51H
FUTURE_BUFF         EQU     52H

                    ORG     00H

                    AJMP    INITIAL

                    ORG 0BH

                    AJMP    TIMING

                    ORG     40H

INITIAL:            MOV     P1,#0FFH
                    MOV     P2,#0FFH
                    MOV     P3,#0FFH
                    MOV     R0,#00H
                    MOV     R1,#00H
                    MOV     R2,#00H
                    MOV     R3,#00H
                    MOV     R4,#00H
                    MOV     R5,#00H
                    MOV     R6,#00H
                    MOV     R7,#00H
                    MOV     TEMP_A,#00H
                    MOV     TEMP_B,#01H
                    MOV     TEMP_C,#02H
                    MOV     TEMP_D,#03H
                    MOV     TEMP_E,#04H
                    MOV     TEMP_F,#05H
                    MOV     TEMP_G,#06H
                    MOV     TEMP_H,#07H
                    MOV     EVEN_POINTER_1,#0FEH
                    MOV     EVEN_POINTER_2,#0FFH
                    MOV     ODD_POINTER_1,#0FFH
                    MOV     ODD_POINTER_2,#0FFH
                    MOV     PAST,#00H
                    MOV     PRESENT,#00H
                    MOV     FUTURE,#00H
                    MOV     TIME_BUF1,#00H
                    MOV     TIME_BUF2,#00H
                    MOV     TIME_BUF3,#00H
                    MOV     DPTR,#0000H
                    MOV     96H,#0AH
                    MOV     TMOD,#01H
                    MOV     TL0,#00H
                    MOV     TH0,#00H
                    SETB    EA
                    SETB    ET0

                    SJMP    MAIN

;.....RECOGNIZED START.....
MAIN:               JB      START_BUTTOM,$
                    MOV     R1,#00H
START:              ACALL   FORWARD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
A1:     MOV   R2,SENSORS
        CJNE  R2,#7FH,A2
        ACALL GET_SENSORS
        ACALL TURN_LEFT
        SJMP  B1
A2:     CJNE  R2,#3FH,A3
        ACALL GET_SENSORS
        ACALL TURN_LEFT
        SJMP  B1
A3:     CJNE  R2,#9FH,A4
        ACALL GET_SENSORS
        ACALL TURN_RIGHT
        SJMP  B1
A4:     CJNE  R2,#0DFH,A1
        ACALL GET_SENSORS
        ACALL TURN_RIGHT
        SJMP  B1
B1:     MOV   R2,SENSORS
        CJNE  R2,#0BFH,B1
        ACALL GET_SENSORS
        INC   R4
        CJNE  R4,#04H,START
        MOV   PAST_BUFF,PAST
    MOV   PRESENT_BUFF,PRESENT
    MOV   FUTURE_BUFF,FUTURE
D1:     ACALL MAKE_TABLE_1
        ACALL MAKE_TABLE_2
        CJNE  R1,#01H,DD1
        SJMP  E1
DD1:    ACALL FORWARD
        ACALL GET_TURN
        CJNE  R1,#01H,D11
        SJMP  E1
D11:    MOV   R2,SENSORS
        CJNE  R2,#0BFH,D11
        ACALL GET_SENSORS
        SJMP  D1
E1:     ACALL STOP
        CLR   P3.0
        MOV   R4,#00H
        MOV   R1,#00H
        MOV   R3,#00H

;.....RECOGNIZING END.....

;----------RECALLING START----------
L1:     JB    RECALL_MODE,$
        NOP
        NOP
        NOP
        NOP
        NOP
        SETB  P3.0

RECALL: ACALL FORWARD
L11:    MOV   R2,SENSORS
        CJNE  R2,#7FH,L12
        ACALL GET_SENSORS
        ACALL TURN_LEFT
        SJMP  L2
L12:    CJNE  R2,#3FH,L13
        ACALL GET_SENSORS
        ACALL TURN_LEFT
        SJMP  L2
L13:    CJNE  R2,#9FH,L14
        ACALL GET_SENSORS
        ACALL TURN_RIGHT
        SJMP  L2
L14:    CJNE  R2,#0DFH,L11
        ACALL GET_SENSORS
        ACALL TURN_RIGHT
        SJMP  L2
L2:     MOV   R2,SENSORS
        CJNE  R2,#0BFH,L2
        ACALL GET_SENSORS
        INC   R4
        CJNE  R4,#04H,RECALL
        MOV   PAST,PRESENT
        MOV   PRESENT,FUTURE
        SJMP  M1
M1:     ACALL MAKE_TABLE_1
        ACALL COMPARE_REM
        LCALL VOTED
        LCALL SOLUTION
        MOV   A,FUTURE
        ANL   A,#1FH
        MOV   TIME_BUF3,A
        MOV   A,FUTURE
        ANL   A,#0E0H
        ORL   A,#1FH
        MOV   R2,A
H1:     CJNE  R2,#7FH,H2
        ACALL TURN_LEFT
        ACALL DIS_CHECK
        SJMP  Q1
H2:     CJNE  R2,#3FH,H3
        ACALL TURN_LEFT
        ACALL DIS_CHECK
        SJMP  Q1
H3:     CJNE  R2,#9FH,H4
        ACALL TURN_RIGHT
        ACALL DIS_CHECK
        SJMP  Q1
H4:     CJNE  R2,#0DFH,H5
        ACALL TURN_RIGHT
        ACALL DIS_CHECK
        SJMP  Q1
H5:     CJNE  R2,#0BFH,H6
        ACALL FORWARD
        ACALL DIS_CHECK
        SJMP  Q1
H6:     CJNE  R2,#0FFH,H7
        ACALL FORWARD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ACALL   DIS_CHECK
                SJMP    Q1
H7:             CJNE    R2,#1FH,Q2
                ACALL   FORWARD
                ACALL   DIS_CHECK

                SJMP    Q1
Q1:             MOV     PAST,PRESENT
                MOV     PRESENT,FUTURE
                SJMP    M1
Q2:             AJMP    E1
;----------RECALLING END----------

;++++++++++++INTERUPT SEVICES+++++++++++++
TIMING:         PUSH    ACC
                INC     TIME_BUF1
                MOV     A,TIME_BUF1
                CJNE    A,#01H,END_INTER
                INC     TIME_BUF2
                MOV     TIME_BUF1,#00H
END_INTER:      POP     ACC
                RETI
;++++++++++++++++++++++++++++++++++++++++

;************SUB ROUTINES******************

;11111111111111111111111111111111111111111111111
FORWARD:        SETB    TR0
                SETB    RY_1
                CLR     RY_2
                SETB    RY_3
                CLR     RY_4
                SETB    LEFT_LED
                SETB    RIGHT_LED
                RET
;11111111111111111111111111111111111111111111111

;22222222222222222222222222222222222222222222222
TURN_LEFT:      SETB    TR0
                SETB    RY_1
                CLR     RY_2
                CLR     RY_3
                SETB    RY_4
                CLR     LEFT_LED
                SETB    RIGHT_LED
                RET
;22222222222222222222222222222222222222222222222

;33333333333333333333333333333333333333333333333
TURN_RIGHT:     SETB    TR0
                CLR     RY_1
                SETB    RY_2
                SETB    RY_3
                CLR     RY_4
                SETB    LEFT_LED
                CLR     RIGHT_LED
                RET
;33333333333333333333333333333333333333333333333

;44444444444444444444444444444444444444444444444
BACKWARD:       SETB    TR0
                CLR     RY_1
                SETB    RY_2
                CLR     RY_3
                SETB    RY_4
                SETB    LEFT_LED
                SETB    RIGHT_LED
                RET
;44444444444444444444444444444444444444444444444

;55555555555555555555555555555555555555555555555
STOP:           SETB    RY_1
                SETB    RY_2
                SETB    RY_3
                SETB    RY_4
                CLR     TR0
                CLR     LEFT_LED
                CLR     RIGHT_LED
                RET
;55555555555555555555555555555555555555555555555

;66666666666666666666666666666666666666666666666
GET_SENSORS:    PUSH    ACC
                ACALL   STOP
                MOV     PAST,PRESENT
                MOV     PRESENT,FUTURE
                MOV     A,R3
                ORL     A,TIME_BUF2
                MOV     FUTURE,A
                MOV     A,R2;SENSORS
                ANL     A,#0E0H
                MOV     R3,A
                MOV     TL0,#00H
                MOV     TH0,#00H
                MOV     TIME_BUF1,#00H
                MOV     TIME_BUF2,#00H
                POP     ACC
                RET
;66666666666666666666666666666666666666666666666

;77777777777777777777777777777777777777777777777
GET_TURN:       PUSH    ACC
C1:             MOV     R2,SENSORS
                CJNE    R2,#7FH,C2
                ACALL   GET_SENSORS
                ACALL   MAKE_TABLE_1
                ACALL   MAKE_TABLE_2
                CJNE    R1,#01,CC1
                SJMP    F1
CC1:            ACALL   TURN_LEFT
                SJMP    F1
C2:             CJNE    R2,#3FH,C3
                ACALL   GET_SENSORS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ACALL   MAKE_TABLE_1
                ACALL   MAKE_TABLE_2
                CJNE    R1,#01,CC2
                SJMP    F1
CC2:            ACALL   TURN_LEFT
                SJMP    F1
C3:             CJNE    R2,#9FH,C4
                ACALL   GET_SENSORS
                ACALL   MAKE_TABLE_1
                ACALL   MAKE_TABLE_2
                CJNE    R1,#01,CC3
                SJMP    F1
CC3:            ACALL   TURN_RIGHT
                SJMP    F1
C4:             CJNE    R2,#0DFH,C5
                ACALL   GET_SENSORS
                ACALL   MAKE_TABLE_1
                ACALL   MAKE_TABLE_2
                CJNE    R1,#01,CC4
                SJMP    F1
CC4:            ACALL   TURN_RIGHT
                SJMP    F1
C5:             JB      START_BUTTOM,C1
                INC     R1
F1:             POP     ACC
                RET
;7777777777777777777777777777777777777777777777777

;8888888888888888888888888888888888888888888888888
DIS_CHECK:      PUSH    ACC
DIS:            MOV     A,TIME_BUF2
                CJNE    A,TIME_BUF3,DIS
                ACALL   STOP
                MOV     TIME_BUF1,#00H
                MOV     TIME_BUF2,#00H
                MOV     TL0,#00H
                MOV     TH0,#00H
                POP     ACC
                RET
;8888888888888888888888888888888888888888888888888

;9999999999999999999999999999999999999999999999999

MAKE_TABLE_1:   PUSH    ACC
                MOV     A,PAST
                MOV     NEURAL_A,#00H
                MOV     NEURAL_B,#00H
                MOV     NEURAL_C,#00H
                MOV     NEURAL_D,#00H
                MOV     NEURAL_E,#00H
                MOV     NEURAL_F,#00H
                MOV     NEURAL_G,#00H
                MOV     NEURAL_H,#00H
                RLC     A       ;BIT 7
                JNC     G1
                SETB    NEURAL_A.7
                SETB    NEURAL_C.7
                SETB    NEURAL_H.7
G1:             RLC     A       ;BIT 6
                JNC     G2
                SETB    NEURAL_A.6
                SETB    NEURAL_B.7
                SETB    NEURAL_C.6
                SETB    NEURAL_H.6
G2:             RLC     A       ;BIT 5
                JNC     G3
                SETB    NEURAL_B.6
                SETB    NEURAL_C.5
                SETB    NEURAL_D.7
                SETB    NEURAL_G.7
                SETB    NEURAL_H.5
G3:             RLC     A       ;BIT 4
                JNC     G4
                SETB    NEURAL_D.6
                SETB    NEURAL_E.7
G4:             RLC     A       ;BIT 3
                JNC     G5
                SETB    NEURAL_A.5
                SETB    NEURAL_B.5
                SETB    NEURAL_D.5
                SETB    NEURAL_E.6
                SETB    NEURAL_F.7
G5:             RLC     A       ;BIT 2
                JNC     G6
                SETB    NEURAL_A.4
                SETB    NEURAL_B.4
                SETB    NEURAL_D.4
                SETB    NEURAL_E.5
                SETB    NEURAL_F.6
                SETB    NEURAL_G.6
G6:             RLC     A       ;BIT 1
                JNC     G7
                SETB    NEURAL_A.3
                SETB    NEURAL_C.4
                SETB    NEURAL_D.3
                SETB    NEURAL_F.5
                SETB    NEURAL_G.5
G7:             RLC     A       ;BIT 0
                JNC     G8
                SETB    NEURAL_A.2
                SETB    NEURAL_B.3
                SETB    NEURAL_C.3
                SETB    NEURAL_G.4
G8:             MOV     A,PRESENT
                RLC     A       ;BIT 7
                JNC     G9
                SETB    NEURAL_E.4
G9:             RLC     A       ;BIT 6
                JNC     G10
                SETB    NEURAL_D.2
                SETB    NEURAL_E.3
                SETB    NEURAL_F.4
G10:            RLC     A       ;BIT 5
                JNC     G11
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SETB    NEURAL_F.3
                SETB    NEURAL_H.4
G11:            RLC     A       ;BIT 4
                JNC     G12
                SETB    NEURAL_F.2
                SETB    NEURAL_G.3
                SETB    NEURAL_H.3
G12:            RLC     A       ;BIT 3
                JNC     G13
                SETB    NEURAL_A.1
                SETB    NEURAL_C.2
                SETB    NEURAL_E.2
                SETB    NEURAL_F.1
                SETB    NEURAL_G.2
                SETB    NEURAL_H.2
G13:            RLC     A       ;BIT 2
                JNC     G14
                SETB    NEURAL_A.0
                SETB    NEURAL_B.2
                SETB    NEURAL_C.1
                SETB    NEURAL_E.1
                SETB    NEURAL_F.0
                SETB    NEURAL_G.1
                SETB    NEURAL_H.1
G14:            RLC     A       ;BIT 1
                JNC     G15
                SETB    NEURAL_B.1
                SETB    NEURAL_C.0
                SETB    NEURAL_D.1
                SETB    NEURAL_E.0
                SETB    NEURAL_G.0
                SETB    NEURAL_H.0
G15:            RLC     A       ;BIT 0
                JNC     G16
                SETB    NEURAL_B.0
                SETB    NEURAL_D.0
G16:            POP     ACC
                RET
;99999999999999999999999999999999999999999999999999

;*10*10*10*10*10*1010*10*10*10*10*10*10*10*10*10*
MAKE_TABLE_2:   PUSH    ACC
                CLR     TEMP_A.3
                CLR     TEMP_B.3
                CLR     TEMP_C.3
                CLR     TEMP_D.3
                CLR     TEMP_E.3
                CLR     TEMP_F.3
                CLR     TEMP_G.3
                CLR     TEMP_H.3
                MOV     A,FUTURE
                RLC     A
                JNC     G17
                SETB    TEMP_A.3
G17:            RLC     A
                JNC     G18
                SETB    TEMP_B.3
G18:            RLC     A
                JNC     G19
                SETB    TEMP_C.3
G19:            RLC     A
                JNC     G20
                SETB    TEMP_D.3
G20:            RLC     A
                JNC     G21
                SETB    TEMP_E.3
G21:            RLC     A
                JNC     G22
                SETB    TEMP_F.3
G22:            RLC     A
                JNC     G23
                SETB    TEMP_G.3
G23:            RLC     A
                JNC     INSERT_REM
                SETB    TEMP_H.3
INSERT_REM:     INC     R5
                ACALL   EVEN
                MOV     A,NEURAL_A
                ACALL   WRITE_EEPROM
                ACALL   EVEN
                MOV     A,NEURAL_B
                ACALL   WRITE_EEPROM
                ACALL   EVEN
                MOV     A,NEURAL_C
                ACALL   WRITE_EEPROM
                ACALL   EVEN
                MOV     A,NEURAL_D
                ACALL   WRITE_EEPROM
                ACALL   EVEN
                MOV     A,NEURAL_E
                ACALL   WRITE_EEPROM
                ACALL   EVEN
                MOV     A,NEURAL_F
                ACALL   WRITE_EEPROM
                ACALL   EVEN
                MOV     A,NEURAL_G
                ACALL   WRITE_EEPROM
                ACALL   EVEN
                MOV     A,NEURAL_H
                ACALL   WRITE_EEPROM
                ACALL   ODD
                MOV     A,TEMP_A
                ACALL   WRITE_EEPROM
                ACALL   ODD
                MOV     A,TEMP_B
                ACALL   WRITE_EEPROM
                ACALL   ODD
                MOV     A,TEMP_C
                ACALL   WRITE_EEPROM
                ACALL   ODD
                MOV     A,TEMP_D
                ACALL   WRITE_EEPROM
                ACALL   ODD
                MOV     A,TEMP_E
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ACALL   WRITE_EEPROM
                ACALL   ODD
                MOV     A,TEMP_F
                ACALL   WRITE_EEPROM
                ACALL   ODD
                MOV     A,TEMP_G
                ACALL   WRITE_EEPROM
                ACALL   ODD
                MOV     A,TEMP_H
                ACALL   WRITE_EEPROM
                ACALL   TEST
                POP     ACC
                RET
;*10*10*10*10*10*1010*10*10*10*10*10*10*10*10*10*10*
TEST:           PUSH    ACC
                CJNE    R5,#01H,TEST1
                SJMP    TEST_END
TEST1:          MOV     A,PAST
        CJNE    A,PAST_BUFF,TEST_END
                MOV     A,PRESENT
        CJNE    A,PRESENT_BUFF,TEST_END
                MOV     A,FUTURE
        CJNE    A,FUTURE_BUFF,TEST_END
                INC     R1
TEST_END:       POP     ACC
                RET

;*11*11*11*11*11*11*11*11*1*11*11*11*11*11*11*
EVEN:           MOV     DPL,EVEN_POINTER_1
                MOV     DPH,EVEN_POINTER_2
                INC     DPTR
                INC     DPTR
                MOV     EVEN_POINTER_1,DPL
                MOV     EVEN_POINTER_2,DPH
                RET
;*11*11*11*11*11*11*11*11*11*11*11*11*11*11*11*

;*12*12*12*12*12*12*12*12*12*12*12*12*12*12*12*
ODD:            MOV     DPL,ODD_POINTER_1
                MOV     DPH,ODD_POINTER_2
                INC     DPTR
                INC     DPTR
                MOV     ODD_POINTER_1,DPL
                MOV     ODD_POINTER_2,DPH
                RET
;*12*12*12*12*12*12*12*12*12*12*12*12*12*12*12*

;*13*13*13*13*13*13*13*13*13*13*13*13*13*13*13*
WRITE_EEPROM:   MOV     96H,#1AH
                MOVX    @DPTR,A
LOOP_1:         MOV     A,96H
                JNB     ACC.1,LOOP_1
                MOV     96H,#0AH
                RET
;*13*13*13*13*13*13*13*13*13*13*13*13*13*13*13*

;*14*14*14*14*14*14*14*14*14*14*14*14*14*14*14*14*
ADD_16:         PUSH    ACC
                MOV     A,DPL
                ADD     A,#10H
                MOV     DPL,A
                MOV     A,DPH
                ADDC    A,#00H
                MOV     DPH,A
                POP     ACC
                RET
;*14*14*14*14*14*14*14*14*14*14*14*14*14*14*14*14*

;*15*15*15*15*15*15*15*15*15*15*15*15*15*15*15*15*

COMPARE_REM:    PUSH    ACC
                MOV     AN_1,#00H
                MOV     BN_1,#00H
                MOV     CN_1,#00H
                MOV     DN_1,#00H
                MOV     EN_1,#00H
                MOV     FN_1,#00H
                MOV     GN_1,#00H
                MOV     HN_1,#00H

LL1A:           MOV     DPTR,#0000H
                MOV     COUNT_1,R5
                SJMP    LL1D
LL1B:           DJNZ    COUNT_1,LL1C
                MOV     DPTR,#0000H
                MOV     NEURAL_BUFF,NEURAL_A
                ACALL   GUESS
                ;
                MOV     AN_1,#00H
                MOV     DPTR,#0000H
                SJMP    LL1CC
LL1BB:          LCALL   ADD_16
                INC     AN_1
LL1CC:          MOV     A,DPL
                CJNE    A,TEMP_0,LL1BB
                MOV     A,DPH
                CJNE    A,TEMP_1,LL1BB
                ;
                SJMP    LL2A
LL1C:           ACALL   ADD_16
                INC     AN_1
LL1D:           MOVX    A,@DPTR
                CJNE    A,NEURAL_A,LL1B
LL2A:           MOV     DPTR,#0002H
                MOV     COUNT_1,R5
                SJMP    LL2D
LL2B:           DJNZ    COUNT_1,LL2C
                MOV     DPTR,#0002H
                MOV     NEURAL_BUFF,NEURAL_B
                ACALL   GUESS
                ;
                MOV     BN_1,#00H
                MOV     DPTR,#0002H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SJMP    LL2CC
LL2BB:          LCALL   ADD_16
                INC     BN_1
LL2CC:          MOV     A,DPL
                CJNE    A,TEMP_0,LL2BB
                MOV     A,DPH
                CJNE    A,TEMP_1,LL2BB
                ;
                SJMP    LL3A
LL2C:           ACALL   ADD_16
                INC     BN_1
LL2D:           MOVX    A,@DPTR
                CJNE    A,NEURAL_B,LL2B
LL3A:           MOV     DPTR,#0004H
                MOV     COUNT_1,R5
                SJMP    LL3D
LL3B:           DJNZ    COUNT_1,LL3C
                MOV     DPTR,#0004H
        MOV     NEURAL_BUFF,NEURAL_C
                ACALL   GUESS
                ;
                MOV     CN_1,#00H
                MOV     DPTR,#0004H
                SJMP  LL3CC
LL3BB:          LCALL   ADD_16
                INC     CN_1
LL3CC:          MOV     A,DPL
                CJNE    A,TEMP_0,LL3BB
                MOV     A,DPH
                CJNE    A,TEMP_1,LL3BB
                ;
                SJMP    LL4A
LL3C:           ACALL   ADD_16
                INC     CN_1
LL3D:           MOVX    A,@DPTR
                CJNE    A,NEURAL_C,LL3B
LL4A:           MOV     DPTR,#0006H
                MOV     COUNT_1,R5
                SJMP    LL4D
LL4B:           DJNZ    COUNT_1,LL4C
                MOV     DPTR,#0006H
        MOV     NEURAL_BUFF,NEURAL_D
                ACALL   GUESS
                ;
                MOV     DN_1,#00H
                MOV     DPTR,#0006H
                SJMP  LL4CC
LL4BB:          LCALL   ADD_16
                INC     DN_1
LL4CC:          MOV     A,DPL
                CJNE    A,TEMP_0,LL4BB
                MOV     A,DPH
                CJNE    A,TEMP_1,LL4BB
                ;
                SJMP    LL5A
LL4C:           ACALL   ADD_16
                INC     DN_1
LL4D:           MOVX    A,@DPTR
                CJNE    A,NEURAL_D,LL4B
LL5A:           MOV     DPTR,#0008H
                MOV     COUNT_1,R5
                SJMP    LL5D
LL5B:           DJNZ    COUNT_1,LL5C
                MOV     DPTR,#0008H
                MOV     NEURAL_BUFF,NEURAL_E
                ACALL   GUESS
                ;
                MOV     EN_1,#00H
                MOV     DPTR,#0008H
                SJMP  LL5CC
LL5BB:          LCALL   ADD_16
                INC     EN_1
LL5CC:          MOV     A,DPL
                CJNE    A,TEMP_0,LL5BB
                MOV     A,DPH
                CJNE    A,TEMP_1,LL5BB
                ;
                SJMP    LL6A
LL5C:           ACALL   ADD_16
                INC     EN_1
LL5D:           MOVX    A,@DPTR
                CJNE    A,NEURAL_E,LL5B
LL6A:           MOV     DPTR,#000AH
                MOV     COUNT_1,R5
                SJMP    LL6D
LL6B:           DJNZ    COUNT_1,LL6C
                MOV     DPTR,#000AH
                MOV     NEURAL_BUFF,NEURAL_F
                ACALL   GUESS
                ;
                MOV     FN_1,#00H
                MOV     DPTR,#000AH
                SJMP  LL6CC
LL6BB:          LCALL   ADD_16
                INC     FN_1
LL6CC:          MOV     A,DPL
                CJNE    A,TEMP_0,LL6BB
                MOV     A,DPH
                CJNE    A,TEMP_1,LL6BB
                ;
                SJMP    LL7A
LL6C:           ACALL   ADD_16
                INC     FN_1
LL6D:           MOVX    A,@DPTR
                CJNE    A,NEURAL_F,LL6B
LL7A:           MOV     DPTR,#000CH
                MOV     COUNT_1,R5
                SJMP    LL7D
LL7B:           DJNZ    COUNT_1,LL7C
                MOV     DPTR,#000CH
                MOV     NEURAL_BUFF,NEURAL_G
                ACALL   GUESS
                ;
                MOV     GN_1,#00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     DPTR,#000CH
                SJMP    LL7CC
LL7BB:          LCALL   ADD_16
                INC     GN_1
LL7CC:          MOV     A,DPL
                CJNE    A,TEMP_0,LL7BB
                MOV     A,DPH
                CJNE    A,TEMP_1,LL7BB
                ;
                SJMP    LL8A
LL7C:           ACALL   ADD_16
                INC     GN_1
LL7D:           MOVX    A,@DPTR
                CJNE    A,NEURAL_G,LL7B
LL8A:           MOV     DPTR,#000EH
                MOV     COUNT_1,R5
                SJMP    LL8D
LL8B:           DJNZ    COUNT_1,LL8C
                MOV     DPTR,#000EH
        MOV     NEURAL_BUFF,NEURAL_H
                ACALL   GUESS
                ;
                MOV     HN_1,#00H
                MOV     DPTR,#000EH
                SJMP    LL8CC
LL8BB:          LCALL   ADD_16
                INC     HN_1
LL8CC:          MOV     A,DPL
                CJNE    A,TEMP_0,LL8BB
                MOV     A,DPH
                CJNE    A,TEMP_1,LL8BB
                ;
                SJMP    LL9A
LL8C:           ACALL   ADD_16
                INC     HN_1
LL8D:           MOVX    A,@DPTR
                CJNE    A,NEURAL_H,LL8B
LL9A:           POP     ACC
                RET
;*15*15*15*15*15*15*15*15*15*15*15*15*15*15*15*15*15*15*15*

;*16*16*16*16*16*16*16*16*16*16*16*16*16*16*16*16*16*16*16*
GUESS:          PUSH    ACC
                MOV     COUNT_1,R5
                MOV     MIN,#0FFH
                MOV     MAX,#00H
                MOV     TEMP_0,#00H
                MOV     TEMP_1,#00H
GUESS_1:        MOV     R6,#08H
                MOVX    A,@DPTR
                XRL     A,NEURAL_BUFF
GUESS_2:        RLC     A
                JNC     GUESS_3
                INC     MAX
GUESS_3:        DJNZ    R6,GUESS_2
                CLR     C
                MOV     A,MIN
                SUBB    A,MAX
                JC      GUESS_4
                MOV     MIN,MAX
                MOV     TEMP_0,DPL
                MOV     TEMP_1,DPH
GUESS_4:        CLR     C
                ACALL   ADD_16
                DJNZ    COUNT_1,GUESS_1
                POP     ACC
                RET
;*16*16*16*16*16*16*16*16*16*16*16*16*16*16*16*16*16*16*16*
;*17*17*17*17*17*17*17*17*17*17*17*17*17*17*17*17*17*17*17*
DELAY2:         MOV     R6,#200
DELAY_11:       MOV     R7,#0E6H
DELAY_22:       NOP
                NOP
                DJNZ    R7,DELAY_22
                DJNZ    R6,DELAY_11
                RET
;*17*17*17*17*17*17*17*17*17*17*17*17*17*17*17*17*17*17*17*
;*18*18*18*18*18*18*18*18*18*18*18*18*18*18*18*18*18*18*18*
VOTE_A:         PUSH    ACC
                MOV     A,AN_1
                CJNE    A,BN_1,VA_1
                INC     AN_2
VA_1:           CJNE    A,CN_1,VA_2
                INC     AN_2
VA_2:           CJNE    A,DN_1,VA_3
                INC     AN_2
VA_3:           CJNE    A,EN_1,VA_4
                INC     AN_2
VA_4:           CJNE    A,FN_1,VA_5
                INC     AN_2
VA_5:           CJNE    A,GN_1,VA_6
                INC     AN_2
VA_6:           CJNE    A,HN_1,VA_7
                INC     AN_2
VA_7:           POP     ACC
                RET
VOTE_B:         PUSH    ACC
                MOV     A,BN_1
                CJNE    A,CN_1,VB_1
                INC     BN_2
VB_1:           CJNE    A,DN_1,VB_2
                INC     BN_2
VB_2:           CJNE    A,EN_1,VB_3
                INC     BN_2
VB_3:           CJNE    A,FN_1,VB_4
                INC     BN_2
VB_4:           CJNE    A,GN_1,VB_5
                INC     BN_2
VB_5:           CJNE    A,HN_1,VB_6
                INC     BN_2
VB_6:           POP     ACC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET                                ;*18*18*18*18*18*18*18*18*18*18*18*18*18*18*18*18*18*18*

VOTE_C: PUSH    ACC                        ;*19*19*19*19*19*19*19*19*19*19*19*19*19*19*19*19*19*19*
        MOV     A,CN_1             VOTED:  MOV     AN_2,#01H
        CJNE    A,DN_1,VC_1                MOV     BN_2,#01H
        INC     CN_2                       MOV     CN_2,#01H
VC_1:   CJNE    A,EN_1,VC_2                MOV     DN_2,#01H
        INC     CN_2                       MOV     EN_2,#01H
VC_2:   CJNE    A,FN_1,VC_3                MOV     FN_2,#01H
        INC     CN_2                       MOV     GN_2,#01H
VC_3:   CJNE    A,GN_1,VC_4                MOV     HN_2,#01H
        INC     CN_2                       ACALL   VOTE_A
VC_4:   CJNE    A,HN_1,VC_5                ACALL   VOTE_B
        INC     CN_2                       ACALL   VOTE_C
VC_5:   POP     ACC                        ACALL   VOTE_D
        RET                                ACALL   VOTE_E
                                           ACALL   VOTE_F
VOTE_D: PUSH    ACC                        ACALL   VOTE_G
        MOV     A,DN_1                     PUSH    ACC
        CJNE    A,EN_1,VD_1                MOV     COUNT_1,#00H
        INC     DN_2                       MOV     R0,#40H   ;AN_2
VD_1:   CJNE    A,FN_1,VD_2                MOV     R7,#08H
        INC     DN_2                       MOV     MAX,#00H
VD_2:   CJNE    A,GN_1,VD_3                SJMP    VOTED3
        INC     DN_2               VOTED1: INC     R0
VD_3:   CJNE    A,HN_1,VD_4        VOTED2: INC     R0
        INC     DN_2               VOTED3: MOV     MIN,@R0
VD_4:   POP     ACC                        MOV     A,MAX
        RET                                SUBB    A,MIN
                                           JNC     VOTED4
VOTE_E: PUSH    ACC                        CLR     C
        MOV     A,EN_1                     MOV     MAX,MIN
        CJNE    A,FN_1,VE_1                INC     R0
        INC     EN_2                       MOV     COUNT_1,@R0
VE_1:   CJNE    A,GN_1,VE_2                DJNZ    R7,VOTED2
        INC     EN_2                       SJMP    VOTED5
VE_2:   CJNE    A,HN_1,VE_3        VOTED4: DJNZ    R7,VOTED1
        INC     EN_2               VOTED5: POP     ACC
VE_3:   POP     ACC                        RET
        RET                                ;*19*19*19*19*19*19*19*19*19*19*19*19*19*19*19*19*19*19*

VOTE_F: PUSH    ACC                        ;*20*20*20*20*20*20*20*20*20*20*20*20*20*20*20*20*20*20*
        MOV     A,FN_1             SOLUTION: PUSH   ACC
        CJNE    A,GN_1,VF_1                MOV     FUTURE,#00H
        INC     FN_2               SOL_1A: MOV     R6,COUNT_1
VF_1:   CJNE    A,HN_1,VF_2                MOV     DPTR,#0000H
        INC     FN_2                       MOV     A,R6
VF_2:   POP     ACC                        CJNE    A,#00H,SOL_1B
        RET                                SJMP    SOL_1C
                                   SOL_1B: LCALL   ADD_16
VOTE_G: PUSH    ACC                        DJNZ    R6,SOL_1B
        MOV     A,GN_1             SOL_1C: INC     DPTR
        CJNE    A,HN_1,VG_1                MOVX    A,@DPTR
        INC     GN_2                       JNB     ACC.3,SOL_2A
VG_1:   POP     ACC                        MOV     A,FUTURE
        RET                                SETB    ACC.7
                                           MOV     FUTURE,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SJMP    SOL_2A
SOL_2A: MOV     R6,COUNT_1
        MOV     DPTR,#0002H
        MOV     A,R6
        CJNE    A,#00H,SOL_2B
        SJMP    SOL_2C
SOL_2B: LCALL   ADD_16
        DJNZ    R6,SOL_2B
SOL_2C: INC     DPTR
        MOVX    A,@DPTR
        JNB     ACC.3,SOL_3A
        MOV     A,FUTURE
        SETB    ACC.6
        MOV     FUTURE,A
        SJMP    SOL_3A
SOL_3A: MOV     R6,COUNT_1
        MOV     DPTR,#0004H
        MOV     A,R6
        CJNE    A,#00H,SOL_3B
        SJMP    SOL_3C
SOL_3B: LCALL   ADD_16
        DJNZ    R6,SOL_3B
SOL_3C: INC     DPTR
        MOVX    A,@DPTR
        JNB     ACC.3,SOL_4A
        MOV     A,FUTURE
        SETB    ACC.5
        MOV     FUTURE,A
        SJMP    SOL_4A
SOL_4A: MOV     R6,COUNT_1
        MOV     DPTR,#0006H
        MOV     A,R6
        CJNE    A,#00H,SOL_4B
        SJMP    SOL_4C
SOL_4B: LCALL   ADD_16
        DJNZ    R6,SOL_4B
SOL_4C: INC     DPTR
        MOVX    A,@DPTR
        JNB     ACC.3,SOL_5A
        MOV     A,FUTURE
        SETB    ACC.4
        MOV     FUTURE,A
        SJMP    SOL_5A
SOL_5A: MOV     R6,COUNT_1
        MOV     DPTR,#0008H
        MOV     A,R6
        CJNE    A,#00H,SOL_5B
        SJMP    SOL_5C
SOL_5B: LCALL   ADD_16
        DJNZ    R6,SOL_5B
SOL_5C: INC     DPTR
        MOVX    A,@DPTR
        JNB     ACC.3,SOL_6A
        MOV     A,FUTURE
        SETB    ACC.3
        MOV     FUTURE,A
        SJMP    SOL_6A
SOL_6A: MOV     R6,COUNT_1
        MOV     DPTR,#000AH
        MOV     A,R6
        CJNE    A,#00H,SOL_6B
        SJMP    SOL_6C
SOL_6B: LCALL   ADD_16
        DJNZ    R6,SOL_6B
SOL_6C: INC     DPTR
        MOVX    A,@DPTR
        JNB     ACC.3,SOL_7A
        MOV     A,FUTURE
        SETB    ACC.2
        MOV     FUTURE,A
        SJMP    SOL_7A
SOL_7A: MOV     R6,COUNT_1
        MOV     DPTR,#000CH
        MOV     A,R6
        CJNE    A,#00H,SOL_7B
        SJMP    SOL_7C
SOL_7B: LCALL   ADD_16
        DJNZ    R6,SOL_7B
SOL_7C: INC     DPTR
        MOVX    A,@DPTR
        JNB     ACC.3,SOL_8A
        MOV     A,FUTURE
        SETB    ACC.1
        MOV     FUTURE,A
        SJMP    SOL_8A
SOL_8A: MOV     R6,COUNT_1
        MOV     DPTR,#000EH
        MOV     A,R6
        CJNE    A,#00H,SOL_8B
        SJMP    SOL_8C
SOL_8B: LCALL   ADD_16
        DJNZ    R6,SOL_8B
SOL_8C: INC     DPTR
        MOVX    A,@DPTR
        JNB     ACC.3,SOL_9A
        MOV     A,FUTURE
        SETB    ACC.0
        MOV     FUTURE,A
        SJMP    SOL_9A
SOL_9A: POP     ACC
        RET
;*20*20*20*20*20*20*20*20*20*20*20*20*20*20*20*20*20*20*20*20*

        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลงได้ด้วยดี ก็ด้วยความสนับสนุนช่วยเหลือและกำลังใจจากหลายๆฝ่ายด้วยกัน โดยการให้ข้อมูล อุปกรณ์สนับสนุน และชี้แนะข้อคิดเห็นต่างๆ ซึ่งมีประโยชน์อย่างมากในการดำเนินการ ไม่ว่าจะเป็นจากพี่ๆ เพื่อนๆ ในภาคอิเล็กทรอนิกส์และภาคอื่นๆ ที่ให้ความช่วยเหลือทุกๆด้าน เพื่อนๆร่วมห้องโปรเจ็คที่ช่วยกันปรึกษาและให้กำลังใจในการทำงาน โดยเฉพาะอย่างยิ่ง ดร.ยุทธนา คิดใจเดียว ที่ให้คำปรึกษาตลอดเวลา และอาจารย์ทุกๆท่านที่ให้คำปรึกษาเป็นอย่างดี

ทางฝ่ายผู้จัดทำจึงขอขอบคุณทุกๆท่านมา ณ โอกาสนี้

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. Robert L.Harvey,"Neural Network principles",Englewood Cliffs,NJ:Prentice-Hall International,c1994.
2. N.K.Bose,P.Liang,"Neural Network fundamentals with graphs,algorithms,and Application",New york,NY;McGraw-Hill,c1996.
3. Martin T.Hagan,Howard B.Demuth,Mark Beale,"Neural Network design",Boston : PWS, c1996.
4. Partic K.Simpson,"Neural Network theory ,technology,and applications", New york: IEEE Technical Activities Board,c1996

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้